

---

# Entwicklung eines Konzepts zur Integration der verhaltenssemantischen Szeneriebeschreibung in hochgenaue Karten

---

## Masterthesis Nr. 839/21

Bearbeiter: Moritz Valentin Berghöfer | 2398341

Betreuer: Felix Glatzki, M. Sc.  
Moritz Lippert, M. Sc.

---



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



FAHRZEUGTECHNIK  
TU DARMSTADT

---

---

Moritz Valentin Berghöfer

Matrikelnummer: 2398341

Studiengang: Master Maschinenbau

Masterthesis Nr. 839/21

Thema: Entwicklung eines Konzepts zur Integration der verhaltenssemantischen Szeneriebeschreibung in hochgenaue Karten

Eingereicht: 8. April 2022

Technische Universität Darmstadt

Fachgebiet Fahrzeugtechnik

Prof. Dr. rer. nat. Hermann Winner

Otto-Berndt-Straße 2

64287 Darmstadt

---



Dieses Dokument wird bereitgestellt von TUprints – Publikationsservice der TU Darmstadt.

<https://tuprints.ulb.tu-darmstadt.de/>

Lizenz: CC-BY 4.0 International

<https://creativecommons.org/licenses/by/4.0/>



---

---

## Erklärung

---

### **Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 und § 23 Abs. 7 APB der TU Darmstadt**

Hiermit versichere ich, Moritz Valentin Berghöfer, die vorliegende Master-Thesis gemäß § 22 Abs. 7 APB TU Darmstadt ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß § 23 Abs. 7 APD TU Darmstadt überein.

---

### **English translation for information purposes only:**

#### **Thesis Statement pursuant to § 22 paragraph 7 and § 23 paragraph 7 of APB TU Darmstadt**

I herewith formally declare that I, Moritz Valentin Berghöfer, have written the submitted thesis independently pursuant to § 22 paragraph 7 of APB TU Darmstadt. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I clearly marked and separately listed all of the literature and all of the other sources, which I employed when producing this academic work, either literally or in content. This thesis has not been handed in or published before in the same or similar form.

I am aware, that in case of an attempt at deception based on plagiarism (§38 Abs. 2 APB), the thesis would be graded with 5,0 and counted as one failed examination attempt. The thesis may only be repeated once.

In the submitted thesis, the written copies and the electronic version for archiving are pursuant to § 23 paragraph 7 of APB TU Darmstadt identical in content.

Matrikelnummer: 2398341

Datum / Date:

Unterschrift / Signature:

08.04.22

M. Berghöfer

---

---

## Zusammenfassung

---

Mit dem Ziel der Verbesserung der Anforderungsdefinition an hochautomatisierte Fahrfunktionen und der damit einhergehenden Reduktion des Testaufwands für den Sicherheitsnachweis dieser Fahrfunktionen wurde am Fachgebiet Fahrzeugtechnik der TU Darmstadt (FZD) die *verhaltenssemantische Szeneriebeschreibung* (BSSD, engl.: *Behavior-Semantic Scenery Description*) entwickelt. Die BSSD definiert ausgehend vom statischen Verkehrsumfeld Grenzen des geforderten Verhaltens eines automatisierten Fahrzeugs in Form eines Verhaltensraums. Dabei wird die Spezifikation der BSSD durch eine topologische Struktur beschrieben. Somit ist für eine Verwendung der BSSD in möglichen praktischen Anwendungen die Verknüpfung dieser mit geometrischen Informationen notwendig. Daraus resultiert die Notwendigkeit der Integration der BSSD in ein hochgenaues Kartenformat.

In dieser Masterthesis wird ein Konzept für eine ganzheitliche Integration der BSSD in das hochgenaue Kartenformat OpenDRIVE entwickelt und umgesetzt. Dazu werden zu Beginn Anforderungen an eine Integration der BSSD in ein hochgenaues Kartenformat definiert. Der Abgleich dieser Anforderungen mit OpenDRIVE zeigt, dass die bestehenden Strukturen von OpenDRIVE für eine Integration der BSSD grundsätzlich geeignet sind, jedoch keine Möglichkeit der Integration des Verhaltensraums bieten.

Auf Basis der benutzerdefinierten Erweiterbarkeit von OpenDRIVE werden drei Konzepte abgeleitet, die eine Integration der BSSD in OpenDRIVE entsprechend der definierten Anforderungen anstreben. Anhand der Definition und Anwendung von Bewertungskriterien wird ein Konzept für die Umsetzung ausgewählt. Die Grundidee des ausgewählten Konzepts besteht in der Integration der BSSD-Segmente, die einen konstanten Verhaltensraum in longitudinaler Richtung repräsentieren, in die Straßen einer OpenDRIVE-Karte.

Als Grundlage für die Umsetzung wird das ausgewählte Konzept ausdetailliert und bezüglich einer Umsetzung bewertet. Daraus ergibt sich ein Konzept, das eine ganzheitliche Integration der BSSD in OpenDRIVE durch eine sequenzielle Abfolge von Schritten realisiert. Die Ausführung der Schritte erfolgt automatisiert, wobei an zwei Stellen die Möglichkeit manueller Eingaben geboten wird, um die Vollständigkeit und Korrektheit der BSSD-Integration sicherzustellen.

Aus der vollständigen Umsetzung des Konzepts in der Programmiersprache *Python* resultiert ein Software-Tool, das eine Integration der BSSD in eine bestehende OpenDRIVE-Karte ermöglicht. Die anschließende Evaluation der umgesetzten Methode ergibt eine Verifikation gegenüber den zu Beginn definierten Anforderungen. Für eine exemplarische Validierung wird die umgesetzte Methode auf zwei repräsentative reale Szenarien angewendet. Dabei zeigt sich, dass eine vollständige und korrekte Integration der BSSD in die OpenDRIVE-Karten der Szenarien mit der umgesetzten Methode erreichbar ist. Die Voraussetzung dafür ist die Ausführung der im Konzept vorgesehenen manuell durchzuführenden Schritte.

Zusammenfassend ermöglicht die in dieser Arbeit umgesetzte Methode mit gewissen Einschränkungen eine vollständige Integration der BSSD in OpenDRIVE. Eine Weiterentwicklung der Methode zur Reduktion manuell durchzuführender Schritte ist möglich.

---

## Inhaltsverzeichnis

---

Zusammenfassung.....	I
Inhaltsverzeichnis.....	II
Formelzeichen- und Indexverzeichnis .....	V
Abkürzungsverzeichnis.....	VI
Abbildungsverzeichnis.....	VII
Tabellenverzeichnis .....	IX
1 Einleitung.....	1
1.1 Motivation.....	2
1.2 Konkretisierung der Aufgabenstellung .....	2
1.3 Methodik des Vorgehens .....	3
2 Die verhaltenssemantische Szeneriebeschreibung.....	5
2.1 Grundbegriffe.....	5
2.2 Allgemeine Erklärung .....	5
2.3 Der Verhaltensraum .....	6
2.3.1 Speed Limit .....	7
2.3.2 Boundary .....	8
2.3.3 Reservation.....	9
2.3.4 Overtake .....	9
2.3.5 Anwendungsbeispiel .....	10
2.4 Definitionsgrenzen .....	11
2.5 Spezifikation .....	12
2.5.1 Das <way>-Element.....	14
2.5.2 Die Elemente des Verhaltensraums .....	16
2.5.3 Das <node>-Element.....	19
3 Das OpenDRIVE-Kartenformat.....	22
3.1 Allgemeine Informationen .....	22
3.2 Grundstruktur .....	22
3.3 Relevante Elemente.....	24
3.3.1 Das <road>-Element.....	25
3.3.2 Das <junction>-Element.....	32
3.3.3 Das <userData>-Element.....	34
4 Anforderungsdefinition & -analyse .....	36
4.1 Ableitung der Anforderungen .....	36
4.2 Analyse der Anforderungsrealisierung mit OpenDRIVE .....	40
5 Konzeptentwicklung .....	42
5.1 Ableitung von Konzepten .....	42
5.1.1 Konzept 1: <lane>-Ebene .....	43

5.1.2	Konzept 2: <road>-Ebene .....	44
5.1.3	Konzept 3: <OpenDRIVE>-Ebene .....	46
5.2	Auswahl eines Konzepts .....	47
5.2.1	Definition von Bewertungskriterien.....	47
5.2.2	Gewichtung der Bewertungskriterien .....	50
5.2.3	Anwendung der Bewertungskriterien .....	51
5.2.4	Nutzwertanalyse.....	55
5.3	Detaillierung des ausgewählten Konzepts .....	57
5.3.1	Ableitung von Schritten .....	58
5.3.2	Umsetzungsbewertung.....	67
6	Implementierung .....	71
6.1	Auswahl der Programmiersprache .....	71
6.2	Dateistruktur.....	71
6.3	Umsetzung der Konzept-Schritte .....	73
6.3.1	Schritt 1: Einlesen und Prüfen der OpenDRIVE-Datei .....	73
6.3.2	Schritt 2: Erstellen der BSSD-Wurzelemente .....	76
6.3.3	Schritt 3: Erstellen der BSSD-Segmente .....	79
6.3.4	Schritt 4: Erstellen der Fahrstreifen-Gruppierungen.....	86
6.3.5	Schritt 5: Erstellen der BSSD-Fahrstreifen.....	86
6.3.6	Schritt 6: Verknüpfen der OpenDRIVE- & BSSD-Fahrstreifen.....	88
6.3.7	Schritt 7: Erstellen der Verhaltensraum-Minimalstruktur .....	89
6.3.8	Schritt 8: Befüllen des Verhaltensattributs Speed Limit.....	89
6.3.9	Schritt 9: Abspeichern der modifizierten OpenDRIVE-Datei .....	98
6.4	Testen der Implementierung .....	98
6.4.1	Vorgehen.....	98
6.4.2	Betrachtung exemplarischer Tests .....	99
7	Evaluation .....	102
7.1	Verifikation .....	102
7.2	Validierung.....	105
7.2.1	Autobahnabschnitt .....	106
7.2.2	Urbane Kreuzung .....	109
8	Gesamtbewertung.....	113
8.1	Bewertung der Integrierbarkeit von BSSD in OpenDRIVE .....	113
8.2	Bewertung der umgesetzten Methode.....	114
8.2.1	Klassifizierung der Befahrbarkeit von OpenDRIVE-Fahrstreifen.....	115
8.2.2	Segmentierung .....	115
8.2.3	Extraktion des Verhaltensattributs Speed Limit .....	117
8.2.4	Ausblick .....	118
9	Fazit und Ausblick .....	120
Anhang	.....	123

---

A.1	Anforderungsliste.....	123
A.2	Konzeptauswahl.....	125
A.3	type-Attribut OpenDRIVE-Fahrstreifen .....	126
A.4	Regeln automatisierte Extraktion BSSD-Segmente.....	127
A.5	Spezifikation der BSSD-Integration .....	128
A.6	Gesamttestfall.....	130
A.7	Zeitplan .....	133
	Literaturverzeichnis .....	134

---

## Formelzeichen- und Indexverzeichnis

---

Lateinische Buchstaben:

<b>Zeichen</b>	<b>Einheit</b>	<b>Bezeichnung</b>
$h$	m	Kartesische Koordinate senkrecht zur $x,y$ -Ebene (Referenzlinien-Koordinatensystem OpenDRIVE)
$s$	m	Kartesische Koordinate tangential zur Referenzlinie (Referenzlinien-Koordinatensystem OpenDRIVE)
$t$	m	Kartesische Koordinate in der $x,y$ -Ebene senkrecht zur Referenzlinie (Referenzlinien-Koordinatensystem OpenDRIVE)
$u$	m	Kartesische Koordinate entlang der Objekt-Längsachse (Lokal-Koordinatensystem OpenDRIVE)
$v$	m	Kartesische Koordinate entlang der Objekt-Querachse (Lokal-Koordinatensystem OpenDRIVE)
$x$	m	Kartesische Koordinate (Inertial-Koordinatensystem OpenDRIVE)
$y$	m	Kartesische Koordinate (Inertial-Koordinatensystem OpenDRIVE)
$z$	m	Kartesische Koordinate (Inertial-Koordinatensystem OpenDRIVE)
$z$	m	Kartesische Koordinate entlang der Objekt-Hochachse (Lokal-Koordinatensystem OpenDRIVE)

Griechische Buchstaben:

<b>Zeichen</b>	<b>Einheit</b>	<b>Bezeichnung</b>
$\alpha$	° bzw. rad	Rollwinkel
$\beta$	° bzw. rad	Nickwinkel
$\gamma$	° bzw. rad	Gierwinkel

---

## Abkürzungsverzeichnis

---

ASAM	Association for Standardisation of Automation and Measuring Systems
BSSD	Behavior-Semantic Scenery Description (dt.: verhaltenssemantische Szeneriebeschreibung)
FGSV	Forschungsgesellschaft für Straßen- und Verkehrswesen
FZD	Fahrzeugtechnik Darmstadt
SAE	Society of Automotive Engineers
StVO	Straßenverkehrs-Ordnung
VzKat	Verkehrszeichenkatalog
XML	Extensible Markup Language

---

## Abbildungsverzeichnis

---

Abbildung 1-1: Methodik des Vorgehens .....	4
Abbildung 2-1: Unterscheidung der Begriffe Verhaltensgrenze und Verhaltensvorgabe .....	6
Abbildung 2-2: Szenerie mit vier atomaren Verhaltensräumen.....	7
Abbildung 2-3: Verhaltensattribute der BSSD und dazugehörige Indikationselemente .....	8
Abbildung 2-4: Anwendungsbeispiel des Verhaltensraums der BSSD .....	10
Abbildung 2-5: Straßenquerschnitt einer Fahrbahn .....	12
Abbildung 2-6: Topologische Repräsentation eines Straßennetzes .....	12
Abbildung 2-7: Hierarchische Struktur der BSSD-Spezifikation .....	13
Abbildung 2-8: Aufteilung eines Wegs in Segmente und Fahrstreifen bei der BSSD .....	16
Abbildung 2-9: Modellierung von Knoten bei der BSSD.....	20
Abbildung 2-10: Unterteilung des BSSD-Streckennetzes aus Abbildung 2-9 in Fahrstreifen .....	21
Abbildung 3-1: Hierarchische Struktur einer OpenDRIVE-Datei .....	23
Abbildung 3-2: Zusammensetzung einer Straße in OpenDRIVE .....	24
Abbildung 3-3: Referenzline und Koordinatensysteme in OpenDRIVE.....	25
Abbildung 3-4: Aufteilung einer Straße in <i>lane sections</i> und Fahrstreifen in OpenDRIVE .....	27
Abbildung 3-5: Hierarchische Struktur des <code>&lt;lane&gt;</code> -Elements in OpenDRIVE .....	27
Abbildung 3-6: Verwendung des <code>&lt;speed&gt;</code> -Elements in OpenDRIVE .....	28
Abbildung 3-7: Szenerie mit Verkehrszeichen 274-80.....	30
Abbildung 3-8: Zylinderförmiges und quaderförmiges Objekt in OpenDRIVE .....	31
Abbildung 3-9: Objekte durch das <code>&lt;repeat&gt;</code> -Element in OpenDRIVE.....	32
Abbildung 3-10: Beispiel einer Kreuzung in OpenDRIVE .....	33
Abbildung 3-11: Beispiel einer direkten Kreuzung in OpenDRIVE .....	35
Abbildung 5-1: Vergleich der Strukturen von OpenDRIVE und BSSD .....	43
Abbildung 5-2: Grundprinzip Konzept 1, <code>&lt;lane&gt;</code> -Ebene .....	43
Abbildung 5-3: Straße mit unterschiedlicher Einteilung von <i>lane sections</i> und BSSD-Segmenten, Beispiel 1.....	44
Abbildung 5-4: Grundprinzip Konzept 2, <code>&lt;road&gt;</code> -Ebene .....	45
Abbildung 5-5: Straße mit unterschiedlicher Einteilung von <i>lane sections</i> und BSSD-Segmenten, Beispiel 2.....	46
Abbildung 5-6: Grundprinzip Konzept 3, <code>&lt;OpenDRIVE&gt;</code> -Ebene .....	47
Abbildung 5-7: Gesamtstruktur der BSSD-Integration in OpenDRIVE .....	58
Abbildung 5-8: Schritte zur Umsetzung von Konzept 2.....	59
Abbildung 5-9: Beispiel Straße mit befahrbaren und nicht-befahrbaren OpenDRIVE-Fahrstreifen	61
Abbildung 5-10: Beispiel Erstellung von BSSD-Fahrstreifen basierend auf OpenDRIVE-Fahrstreifen .....	63
Abbildung 5-11: Schritte und Algorithmen für eine ideale Realisierung von Konzept 2.....	67
Abbildung 5-12: Schritte und Algorithmen für eine umsetzbare Realisierung von Konzept 2.....	70
Abbildung 6-1: Struktur des Software-Tools.....	72
Abbildung 6-2: Funktionsschaubild Schritt 1 „Einlesen & Prüfen der OpenDRIVE-Datei“ .....	73
Abbildung 6-3: Gesamtschaubild der Implementierung .....	74
Abbildung 6-4: Funktionsschaubild Schritt 2 „Erstellen der BSSD-Segmente“ .....	76
Abbildung 6-5: Beispiel befahrbarer und nicht-befahrbarer Fahrradweg.....	77
Abbildung 6-6: Konsole bei Ausführung des Algorithmus A.2 .....	78

Abbildung 6-7: Funktionsschaubild Schritt 3 „Einlesen & Prüfen der OpenDRIVE-Datei“ .....	79
Abbildung 6-8: Konsole bei Ausführung des Algorithmus A.4 .....	85
Abbildung 6-9: Funktionsschaubild Schritt 5 „Erstellen der BSSD-Fahrstreifen“ .....	87
Abbildung 6-10: Funktionsschaubild Schritt 6 „Suche von zugehörigen OpenDRIVE-Fahrstreifen“ .....	88
Abbildung 6-11: Funktionsschaubild Schritt 8 „Befüllen des Verhaltensattributs <i>Speed Limit</i> “ .....	90
Abbildung 6-12: Zusammenhang zwischen vorgesehener Fahrtrichtung und Referenzrichtung .....	91
Abbildung 6-13: Festlegung der Ausprägung des Verhaltensattributs <i>Speed Limit</i> .....	92
Abbildung 6-14: Trennung eines BSSD-Fahrstreifens vom Gegenverkehr, Kriterium 1 .....	93
Abbildung 6-15: Trennung eines BSSD-Fahrstreifens vom Gegenverkehr, Kriterium 2 .....	93
Abbildung 6-16: Klassifizierung Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen anhand der Nachbarfahrstreifen .....	94
Abbildung 6-17: Objekt innerhalb eines OpenDRIVE-Fahrstreifens .....	96
Abbildung 6-18: Klassifizierung Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen anhand von Objekten .....	98
Abbildung 6-19: Vorgehen zur Durchführung eines Tests .....	99
Abbildung 6-20: Szenerie Test 1 .....	100
Abbildung 6-21: Szenerie und erwartetes Ergebnis Test 2 .....	100
Abbildung 7-1: Übersicht der Szenerie des Gesamttestfalls .....	102
Abbildung 7-2: Szenerie Straße 9, Gesamttestfall .....	103
Abbildung 7-3: Übersicht Abschnitt A9 .....	106
Abbildung 7-4: Detailaufnahme Abschnitt A9 .....	106
Abbildung 7-5: Segmentierung A9, Beispiel 1 .....	108
Abbildung 7-6: Segmentierung A9, Beispiel 2 .....	108
Abbildung 7-7: Übersicht Ausschnitt Griesheim .....	110
Abbildung 7-8: Detailaufnahme Ausschnitt Griesheim .....	110
Abbildung 7-9: Beispiel <i>type</i> -Attribut, Griesheim .....	111
Abbildung 7-10: Beispiel Segmentierung, Griesheim .....	111
Abbildung A-1: UML-Diagramm der BSSD-Integration .....	128
Abbildung A-2: Verhaltensraum-Minimalstruktur .....	128
Abbildung A-3: Szenerie Straßen 0 und 1, Gesamttestfall .....	130
Abbildung A-4: Szenerie Straße 23, Gesamttestfall .....	130
Abbildung A-5: Zeitplan .....	133

---

## Tabellenverzeichnis

---

Tabelle 2-1: Attribute <longitudinal>-Element .....	19
Tabelle 2-2: Attribute <reservation>-Element.....	19
Tabelle 2-3: <connection>- und <laneLink>-Elemente Knoten 4 aus Abbildung 2-10.....	21
Tabelle 3-1: Attribute <signal>-Element für Szenerie in Abbildung 3-7.....	30
Tabelle 3-2: <connection>- und <laneLink>-Elemente Kreuzung aus Abbildung 3-10.....	34
Tabelle 5-1: Paarvergleich der Bewertungskriterien .....	51
Tabelle 5-2: Finale Gewichtung der Bewertungskriterien.....	51
Tabelle 5-3: Anwendung der Bewertungskriterien auf abgeleitete Konzepte .....	55
Tabelle 5-4: Nutzwertanalyse zur Auswahl eines Konzepts.....	56
Tabelle 5-5: <i>lane sections</i> und OpenDRIVE-Fahrstreifen basierend auf Abbildung 5-10.....	64
Tabelle 5-6: BSSD-Segmente und -Fahrstreifen basierend auf Abbildung 5-10.....	64
Tabelle 5-7: Verknüpfung OpenDRIVE- und BSSD-Fahrstreifen basierend auf Abbildung 5-10 ...	65
Tabelle 6-1: Ausprägungen Verhaltensattribut <i>Speed Limit</i> für Szenerie aus Abbildung 6-13.....	92
Tabelle 6-2: Datenstruktur „Objekt-Daten“ .....	95
Tabelle 6-3: Werte „Objekt-Daten“ für Objekt aus Abbildung 6-17.....	96
Tabelle 6-4: Erwartetes Ergebnis Test 1 .....	100
Tabelle 7-1: Kennzahlen Stufe 1, Validierung A9.....	107
Tabelle 7-2: Kennzahlen Stufe 2, Validierung A9.....	107
Tabelle 7-3: Kennzahlen Stufe 1, Validierung Griesheim.....	110
Tabelle 7-4: Kennzahlen Stufe 2, Validierung Griesheim.....	110
Tabelle A-1: Anforderungen an hochgenaues Kartenformat für Vollständigkeit der BSSD- Informationen .....	123
Tabelle A-2: Paarvergleich der Bewertungskriterien, BSSD-Entwickler 1.....	125
Tabelle A-3: Paarvergleich der Bewertungskriterien, BSSD-Entwickler 2.....	125
Tabelle A-4: Nutzwertanalyse, Bewertung anhand Schulnotenskala .....	125
Tabelle A-5: Klassifizierung Befahrbarkeit OpenDRIVE-Fahrstreifen anhand <i>type</i> -Attribut .....	126
Tabelle A-6: Regeln für automatisierte Extraktion von BSSD-Segmenten.....	127
Tabelle A-7: Spezifikation der Attribute der XML-Elemente der BSSD-Integration .....	129
Tabelle A-8: Zusammenhang zwischen Szenerie und zu testenden Funktionen, Gesamttestfall ....	131

---

# 1 Einleitung

---

Im Mai 2021 wurde im Deutschen Bundestag das *Gesetz zum autonomen Fahren*<sup>1</sup> beschlossen. Somit ist Deutschland das erste Land weltweit, in dem die Teilnahme von fahrerlosen Fahrzeugen am öffentlichen Straßenverkehr möglich ist.<sup>2</sup> Dieses Gesetz ist ein Resultat der seit der Einführung in den 70er Jahren an Dynamik und Funktionsvielfalt zunehmenden Entwicklung von Fahrerassistenzsystemen. Mittlerweile stellen Fahrerassistenzsysteme einen wesentlichen Bestandteil der Standardausstattung vieler Fahrzeuge dar. Wie das Gesetz zum autonomen Fahren vermuten lässt, liegt der Entwicklungsfokus von Fahrerassistenzsystemen heutzutage in der Steigerung des Automatisierungsgrades, wobei eine Hauptmotivation dafür die Erhöhung der Verkehrssicherheit ist.<sup>3a</sup> Aktuelles Ziel ist die Einführung von Fahrzeugen mit der Automatisierungsstufe drei, im deutschsprachigen als *Hochautomatisierung* bezeichnet, bei der nach Definition der *Society of Automotive Engineers* (SAE) das Fahrzeug in festgelegten Anwendungsfällen selbständig fährt und sich der Fahrer kurzfristig von seiner Fahraufgabe abwenden darf.<sup>4</sup>

Einen essenziellen Bestandteil für die Erprobung und die Einführung von höheren Automatisierungsstufen, insbesondere in urbanen Verkehrsbereichen, stellen nach aktuellem Stand der Wissenschaft hochgenaue Karten dar.<sup>5,6</sup> Dies zeigt sich auch durch die Tatsache, dass in der Vergangenheit hochgenaue Karten in renommierten Projekten zum automatisierten Fahren<sup>7,8,9,10</sup> eine entscheidende Rolle gespielt haben. Im Gegensatz zu den in Navigationssystemen verwendeten konventionellen Karten, welche die Topologie des Straßennetzes und den geometrischen Verlauf der Straße beinhalten, verfügen hochgenaue Karten zusätzlich über zentimetergenaue Daten der Positionen und Begrenzungen der einzelnen Fahrstreifen. Weiterhin sind Informationen über Objekte der Verkehrsinfrastruktur, wie Tunnel, Brücken oder Lichtsignalanlagen, und regelvorgebende Elemente, wie beispielsweise Verkehrszeichen für Geschwindigkeitsbegrenzungen, in den hochgenauen Karten enthalten.<sup>3b</sup> Mit diesen Eigenschaften erfüllen hochgenaue Karten verschiedene Aufgaben im Kontext eines automatisierten Fahrzeugs. Einerseits unterstützen hochgenaue Karten die Fahrzeugsensorik bei der Erstellung einer semantischen Karte der Umgebung. Andererseits kompensieren hochgenaue Karten Sensorungenauigkeiten und liefern Informationen über Bereiche, die aufgrund von begrenzter Sensorreichweite oder Verdeckung, nicht von der Sensorik erfassbar sind. Weiterhin sind hochgenaue Karten notwendig für virtuelle Fahrversuche, die erhebliche Vorteile gegenüber realen Fahrversuchen hinsichtlich

---

<sup>1</sup> Deutscher Bundestag: Gesetz zum autonomen Fahren (2021).

<sup>2</sup> BMVI: Gesetz zum autonomen Fahren tritt in Kraft (2021).

<sup>3</sup> Bengler, K. et al.: Fahrerassistenzsysteme und Automatisiertes Fahren (2021), a: 1011–1012; b: 1035–1036.

<sup>4</sup> SAE International: SAE International Standard J3016 (2014).

<sup>5</sup> Poggenhans, F. et al.: Lanelet2: A high-definition map framework for the future of automated driving (2018), S. 1672.

<sup>6</sup> Althoff, M. et al.: Automatic Conversion of Road Networks from OpenDRIVE to Lanelets (2018), S. 157.

<sup>7</sup> Wille, J. M. et al.: Stadtpilot: Driving autonomously on Braunschweig's inner ring road (2010).

<sup>8</sup> Ziegler, J. et al.: Making Bertha Drive — An Autonomous Journey on a Historic Route (2014).

<sup>9</sup> Aeberhard, M. et al.: Experience, Results and Lessons Learned from Automated Driving on Germany's Highways (2015).

<sup>10</sup> Tas, O. S. et al.: Making Bertha Cooperate—Team AnnieWAY's Entry to the 2016 Grand Cooperative Driving Challenge (2018).

---

Reproduzierbarkeit, Flexibilität und Aufwandsreduktion bei der Absicherung von höheren Automatisierungsstufen aufweisen.<sup>5,6,11</sup>

## 1.1 Motivation

Innerhalb der Forschungsprojekte PRORETA 5<sup>12</sup> und UNICARagil<sup>13</sup> wird am Fachgebiet Fahrzeugtechnik der TU Darmstadt (FZD) am automatisierten Fahren für den urbanen Verkehr geforscht. Dabei wurde die sogenannte *verhaltenssemantische Szeneriebeschreibung* (BSSD, engl.: *Behavior-Semantic Scenery Description*) entwickelt, die ausgehend vom statischen Verkehrsumfeld Grenzen des geforderten Verhaltens eines automatisierten Fahrzeugs definiert. Die Extraktion der für das Verhalten des Fahrzeugs relevanten Informationen aus dem statischen Verkehrsumfeld verbessert die Definition von Anforderungen an hochautomatisierte Fahrfunktionen. Dies ist für den Sicherheitsnachweis der hochautomatisierten Fahrfunktionen und dem damit einhergehenden Testaufwand von Vorteil.<sup>14</sup>

Für die BSSD wird aktuell eine hierarchisch strukturierte Spezifikation in Anlehnung an das hochgenaue Kartenformat OpenDRIVE<sup>15</sup> entwickelt. OpenDRIVE ist ein in Industrie und Forschung verbreiteter offener Standard zur Beschreibung von hochgenauen Karten im Kontext des automatisierten Fahrens. Da die Geometrie des statischen Verkehrsumfelds keinen Einfluss auf die BSSD hat, ist die Spezifikation der BSSD in topologischer Form definiert und berücksichtigt folglich, im Gegensatz zu einem hochgenauen Kartenformat wie OpenDRIVE, keine geometrischen Eigenschaften des Straßennetzes und der darin enthaltenen Fahrstreifen. Damit die Informationen der BSSD an einer definierten Position eines Straßennetzes zugänglich sind, ist eine Integration der BSSD in ein hochgenaues Kartenformat, wie beispielsweise OpenDRIVE, notwendig. Somit wäre es möglich, Anforderungen an das automatisierte Fahrzeug direkt aus der hochgenauen Karte abzuleiten. Für ein automatisiertes Fahrzeug bestände daher die Möglichkeit, ausgehend von den eigenen technischen Fähigkeiten, direkt im Betrieb zu überprüfen, welche Straßen befahrbar bzw. nicht befahrbar sind. Dies würde insbesondere die schrittweise Einführung von automatisierten Fahrzeugen in den Straßenverkehr unterstützen. Weiterhin würde eine Integration der BSSD in das OpenDRIVE-Kartenformat zur Kompatibilität und Verbreitung der BSSD beitragen.

## 1.2 Konkretisierung der Aufgabenstellung

In dieser Masterthesis soll ein Konzept für die Integration der BSSD in das OpenDRIVE-Kartenformat entwickelt, bezüglich einer Umsetzung bewertet und darauf aufbauend zu Teilen implementiert

---

<sup>11</sup> Hakuli, S.; Krug, M.: Virtuelle Integration (2015), S. 126.

<sup>12</sup> FZD: PRORETA 5.

<sup>13</sup> BMBF: UNICARagil (2022).

<sup>14</sup> Glatzki, F. et al.: Behavior-Semantic Scenery Description (BSSD) (2021), S. 667.

<sup>15</sup> ASAM e.V.: ASAM OpenDRIVE® (2021).

---

werden. Idealerweise ermöglicht das umgesetzte Konzept eine vollständige Integration der BSSD in eine vorhandene OpenDRIVE-Karte.

Hierfür ist zunächst eine Einarbeitung in die beiden Themenbereiche BSSD und OpenDRIVE gefordert. Für das übergeordnete Verständnis dieser Themenbereiche ist weiterhin eine Literaturrecherche und Einarbeitung in das Themengebiet der hochgenauen Karten für automatisiertes Fahren notwendig. Nach der Einarbeitung sollen ausgehend von der BSSD allgemeine Anforderungen definiert werden, deren Erfüllung für die gesamtheitliche Integration der BSSD in ein hochgenaues Kartenformat notwendig ist. Ausgehend von den formulierten Anforderungen ist gefordert, das OpenDRIVE-Format bezüglich einer Integrierbarkeit der BSSD zu analysieren. Das Ziel der Analyse besteht in der Feststellung der zur Integration der BSSD bereits vorhandenen bzw. fehlenden Strukturen und Funktionalitäten in OpenDRIVE.

Auf Basis der vorangegangenen Analyse sollen notwendige Schritte und Maßnahmen abgeleitet werden, um eine Integration der BSSD in das OpenDRIVE-Kartenformat zu ermöglichen. Die abgeleiteten Schritte sollen weiterhin bezüglich ihrer Realisierbarkeit und ihres Umsetzungsaufwands beurteilt werden. Aus den erarbeiteten Schritten und der Umsetzungsbeurteilung folgend, ist das beispielhafte Implementieren und Testen einzelner Schritte in Form von Software-Bausteinen gefordert. Die Arbeit soll anhand der aus den zuvor beschriebenen Schritten resultierenden Erkenntnissen in Form einer ganzheitlichen Analyse und Bewertung bezüglich der Integrierbarkeit von BSSD in OpenDRIVE beendet werden.

### 1.3 Methodik des Vorgehens

Die für die Bearbeitung dieser Masterthesis erarbeitete Methodik ist in Abbildung 1-1 dargestellt. Sie besteht aus fünf übergeordneten Themenblöcken, die wiederum einzelne Schritte beinhalten.

Der erste Themenblock **Literatur & Einarbeitung** beinhaltet einerseits eine Einarbeitung in die BSSD sowie das OpenDRIVE-Format. Dies wird vor allem über die den beiden Themenfeldern zugehörigen Spezifikationen<sup>16,17</sup>, aber auch durch zugehörige wissenschaftliche Veröffentlichungen realisiert. Andererseits wird zum gesamtheitlichen Verständnis der Aufgabenstellung eine Literaturrecherche bezüglich der Motivation und Anwendung von hochgenauen Karten im Kontext des automatisierten Fahrens durchgeführt. Das aus dem Block Literatur & Einarbeitung resultierende Wissen wird als Grundlage dieser Arbeit in den Kapiteln 2 und 3 beschrieben.

Im zweiten Themenblock **Anforderungsdefinition & -analyse** werden basierend auf der Einarbeitung in die BSSD und das Themenfeld der hochgenauen Karten zunächst Anforderungen an eine Integration der BSSD in ein hochgenaues Kartenformat definiert. Daraufhin wird durch den Abgleich dieser Anforderungen mit dem OpenDRIVE-Kartenformat analysiert, welche Strukturen in OpenDRIVE für eine mögliche Integration der BSSD bereits vorhanden, möglicherweise in

---

<sup>16</sup> Glatzki, F.; Lippert, M.: Behavior-Semantic Scenery Description v0.4 (unveröffentlicht) (2021).

<sup>17</sup> ASAM e.V.: Specification ASAM OpenDRIVE, Version 1.7.0 (2021).

abgewandelter Form vorhanden oder nicht vorhanden sind. Die beiden Schritte des Themenblocks Anforderungsdefinition & -analyse sind in Kapitel 4 beschrieben.

Im Themenblock **Konzeptentwicklung** werden in einem ersten Schritt anhand der Ergebnisse der vorigen Analyse und unter Berücksichtigung der Funktionalitäten von OpenDRIVE mehrere Konzepte für eine Integration der BSSD in OpenDRIVE abgeleitet. Anschließend werden Bewertungskriterien definiert anhand derer ein Vergleich der Konzepte möglich ist. Daraufhin wird das Konzept mit dem höchsten Gesamtnutzen ausgewählt und in einzelne zur Umsetzung des Konzepts notwendige Teilschritte zerlegt, die zunächst bezüglich einer Umsetzung bewertet werden. Alle zum Themenblock Konzeptentwicklung zugehörigen Schritte sind in Kapitel 5 dokumentiert.

Anhand der zuvor erfolgten Bewertung werden im nächsten Schritt im Themenblock **Implementierung** einzelne Komponenten des umzusetzenden Konzepts ausgewählt, die beispielhaft als Software-Bausteine implementiert werden. Zur Überprüfung der korrekten Funktionalität der implementierten Komponenten werden einzelne Testfälle definiert und angewendet, sodass ein iterativer Implementierungsprozess resultiert. Der Themenblock **Implementierung** wird in Kapitel 6 beschrieben.

Im letzten Themenblock **Evaluation** wird in Kapitel 7 anhand der zu Beginn definierten Anforderungen sowie der Betrachtung von realen Szenarien eine Verifikation und Validierung der Implementierung durchgeführt. Anhand der daraus resultierenden Ergebnisse, den Testergebnissen der implementierten Komponenten sowie der während der Durchführung der Arbeit gewonnenen Erkenntnisse wird in Kapitel 8 eine gesamtheitliche Bewertung der Integrierbarkeit von BSSD in OpenDRIVE vorgenommen. Die Arbeit wird mit einem Fazit sowie einem Ausblick in Kapitel 9 beendet.

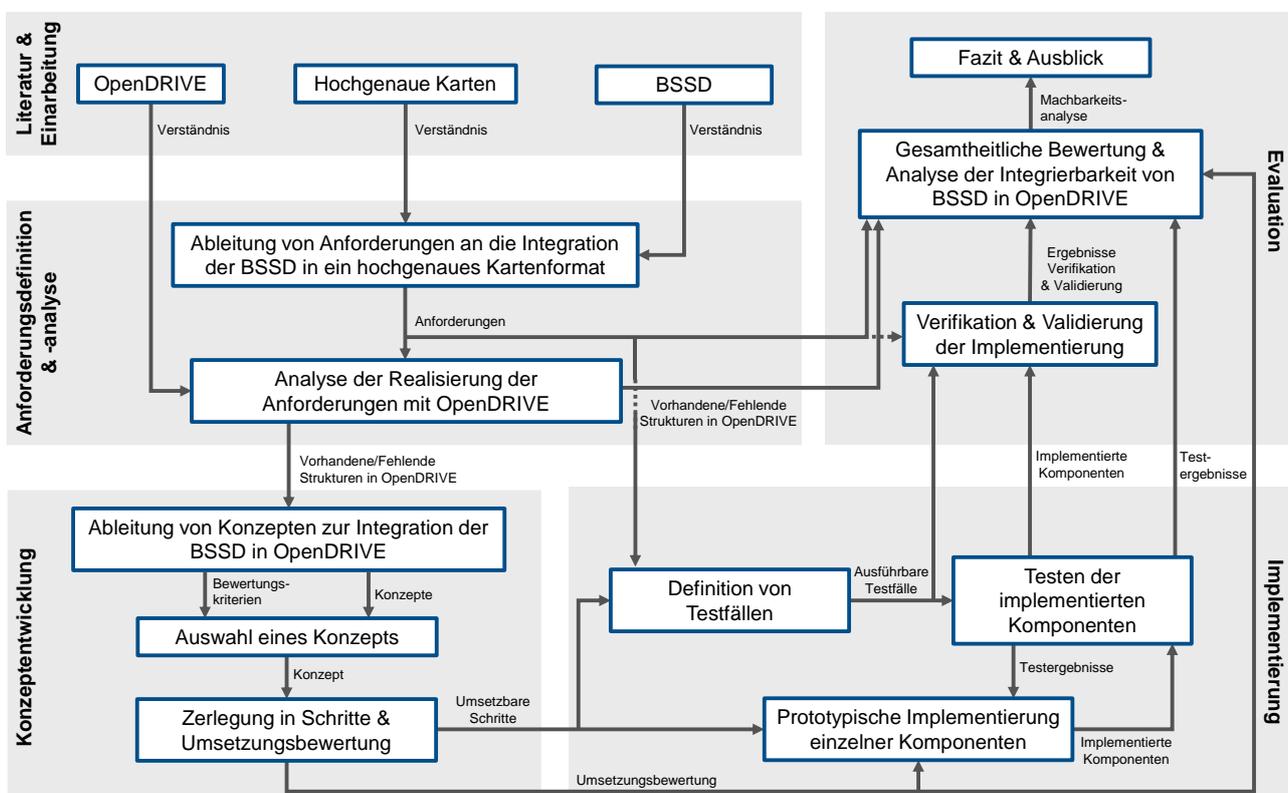


Abbildung 1-1: Methodik des Vorgehens

---

## 2 Die verhaltenssemantische Szeneriebeschreibung

---

In diesem Kapitel wird die bei FZD entwickelte BSSD erläutert. Dazu werden in Abschnitt 2.1 Grundbegriffe definiert, die für das Verständnis der BSSD notwendig sind. Ausgehend von diesen Grundbegriffen wird die BSSD allgemein in Abschnitt 2.2 erklärt. Der inhaltliche Kern der BSSD wird durch den Verhaltensraum gebildet, der in Abschnitt 2.3 beschrieben wird. Die Definitionsgrenzen der BSSD werden in Abschnitt 2.4 festgelegt. In Abschnitt 2.5 wird eine aktuell in Entwicklung befindliche Spezifikation der BSSD erläutert.

### 2.1 Grundbegriffe

BSSD ist die englischsprachige Abkürzung des Begriffs *verhaltenssemantische Szeneriebeschreibung*. Für das Verständnis dieser Beschreibung werden daher zunächst die Begriffe *Szenerie*, *Verhalten* und *Semantik* im Kontext dieser Arbeit erklärt.

Eine **Szenerie** im Kontext des automatisierten Fahrens beinhaltet nach Ulbrich et al.<sup>18</sup> alle Aspekte einer Szene, die räumlich stationär sind. Dabei handelt es sich im Detail um das Fahrstreifennetz, stationäre Elemente (z.B. Bordsteine, Verkehrsschilder oder Lichtsignalanlagen), vertikale Elevationen sowie weitere Umweltbedingungen. Eine Szenerie ist einer von vier Bestandteilen einer Szene. Letztere besteht weiterhin aus dynamischen Elementen, der Selbstrepräsentation aller Akteure und Beobachter sowie der Verknüpfung der drei zuvor genannten Bestandteile.<sup>18</sup>

Das **Verhalten** eines Fahrzeugs wird nach Lippert et al.<sup>19</sup> durch seine Bewegung sowie die Interaktion mit der Umwelt und anderen Verkehrsteilnehmern repräsentiert.

Die **Semantik** beschreibt allgemein einen Prozess, bei dem einer Information eine Bedeutung zugeordnet wird.<sup>20</sup> Im Kontext der BSSD entsprechen diese Informationen primär Verkehrsregeln, deren Auswirkungen auf die Verhaltensgrenzen eines Fahrzeugs abgeleitet werden.

### 2.2 Allgemeine Erklärung

Die BSSD ist eine Beschreibung, die darauf abzielt, das von einem Fahrzeug in einer bestimmten Situation geforderte Verhalten direkt mit der umgebenden Szenerie zu verknüpfen. Folglich nimmt sich die BSSD dem Problem an, den Schritt bzw. die Schritte der Interpretation, die zwischen der das Fahrzeug umgebenden Szenerie einerseits und dem von einem Fahrzeug geforderte Verhalten andererseits liegen, durchzuführen (vgl. Prozess der Semantik, Abschnitt 2.1). Nach aktuellem Stand der Literatur werden diese beiden Aspekte stets getrennt betrachtet.<sup>21a</sup>

---

<sup>18</sup> Ulbrich, S. et al.: Definition der Begriffe Szene, Situation und Szenario für das automatisierte Fahren (2015), S. 2–4.

<sup>19</sup> Lippert, M. et al.: Definition von Bestehens-/Versagenskriterien (2019), S. 2.

<sup>20</sup> Duden Online: Definition Semantik (2021).

<sup>21</sup> Glatzki, F. et al.: Behavior-Semantic Scenery Description (BSSD) (2021), a: S. 669; b: S. 669–670; c: S. 670–671.

Die BSSD beschreibt die Verhaltensgrenzen eines in einer bestimmten Szenerie befindlichen Fahrzeugs. Eine Verhaltensgrenze beschreibt eine Menge möglicher Verhaltensweisen, innerhalb derer das Fahrzeug die Möglichkeit besitzt, eigenständige Verhaltensentscheidungen zu treffen. Dies ist zu unterscheiden von einer Verhaltensvorgabe, die eine explizite Verhaltensweise fordert. Zur Verdeutlichung des Unterschieds der Begriffe Verhaltensgrenze und Verhaltensvorgabe sind in Abbildung 2-1 das Beispiel einer roten Lichtsignalanlage sowie die daraus resultierenden Einschränkungen dargestellt. Die Verhaltensvorgabe fordert im dargestellten Beispiel, dass das Fahrzeug bei roter Lichtsignalanlage an der Haltelinie stehen bleiben muss. Demnach bleibt dem Fahrzeug nur eine mögliche Verhaltensweise: Es muss bis zur Haltelinie vordringen und anschließend an dieser anhalten (Längerer Pfeil in Abbildung 2-1). Die Verhaltensgrenze dagegen fordert im dargestellten Beispiel, dass das Fahrzeug die Haltelinie bei roter Lichtsignalanlage nicht überfahren darf. Nach dieser Verhaltensgrenze ergeben sich dem Fahrzeug somit mehrere mögliche Verhaltensweisen. Es besteht beispielsweise die Möglichkeit, dass sich das Fahrzeug wie bei der Verhaltensvorgabe verhält (Vordringen und halten). Weiterhin wäre ein langsames Annähern des Fahrzeugs an die Haltelinie möglich, sodass das Fahrzeug nach einem Wechsel der Lichtsignalanlage auf Grün die Haltelinie ohne einen Halt überfährt (kürzerer Pfeil in Abbildung 2-1).<sup>21a</sup>

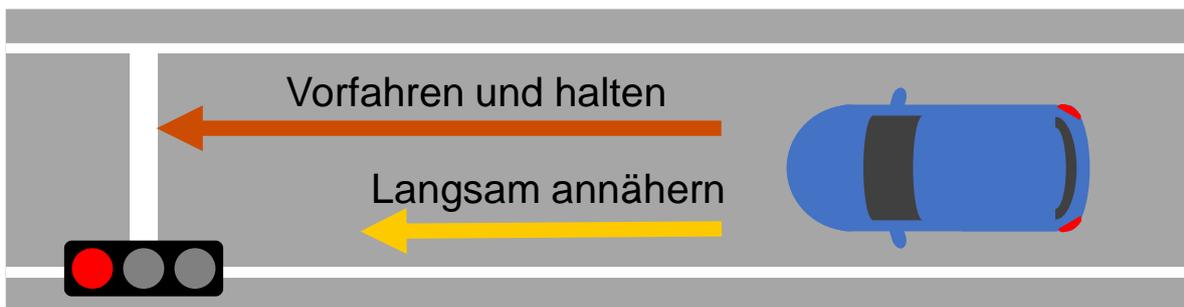


Abbildung 2-1: Unterscheidung der Begriffe Verhaltensgrenze und Verhaltensvorgabe<sup>22</sup>

### 2.3 Der Verhaltensraum

Die Menge aller möglichen Verhaltensweisen, die sich einem in einer bestimmten Szenerie befindlichen Fahrzeug aufgrund der Verhaltensgrenzen ergeben, werden im Kontext der BSSD als *Verhaltensraum* bezeichnet. Innerhalb dieses Verhaltensraums bietet sich dem Fahrzeug die Möglichkeit, ohne weitere Vorgaben zu handeln. Der Verhaltensraum leitet sich aus der Szenerie sowie anhand der in der Szenerie geltenden Verkehrsregeln ab. Im Fall der BSSD wird die deutsche Straßenverkehrs-Ordnung<sup>23</sup> (StVO) als Grundlage für die innerhalb einer Szenerie geltenden Verkehrsregeln verwendet. Neben den länderspezifischen Verkehrsregeln beeinflussen allgemein akzeptierte Regeln zum Fahren im Straßenverkehr, wie sie beispielsweise von Shalev-Shwartz et al.<sup>24</sup> formuliert werden, den Verhaltensraum. Eine solche Regel lautet beispielsweise: „Wenn du einen Unfall verhindern kannst, ohne dabei einen anderen Unfall auszulösen, dann musst du das tun“<sup>24, 21a</sup>.

<sup>22</sup> Eigene Darstellung basierend auf einer internen Präsentation von Moritz Lippert (FZD)

<sup>23</sup> BMJV: Straßenverkehrs-Ordnung (StVO) (2013).

<sup>24</sup> Shalev-Shwartz, S. et al.: On a Formal Model of Safe and Scalable Self-driving Cars (2017), S. 6.

Der Verhaltensraum der BSSD besteht aus vier Dimensionen, die durch die sogenannten *Verhaltensattribute* repräsentiert werden. Das Ziel der Verhaltensattribute besteht darin, alle möglicherweise auftretenden Regelarten zu berücksichtigen und soweit zu abstrahieren, dass das eigentliche die Regel festlegende Element (z.B. ein Verkehrsschild) nicht im Verhaltensattribut repräsentiert ist. Elemente, die eine Regel festlegen, welche mindestens ein Verhaltensattribut beeinflusst, werden im Folgenden als *Indikationselemente* bezeichnet. Jedes Verhaltensattribut hat einzelne untergeordnete Eigenschaften, die den Zustand dieses Verhaltensattributs bestimmen. Die Gesamtheit der vier Verhaltensattribute mit jeweils konkret festgelegten Eigenschaften bilden den sogenannten *atomaren Verhaltensraum*. Somit ist der atomare Verhaltensraum ein Bestandteil des Straßennetzes, in dem sich die Verhaltensattribute nicht ändern. Im Großteil der Fälle entspricht dies einem in longitudinaler Richtung abgegrenzten Teil eines Fahrstreifens.<sup>21b</sup>

Zur Verdeutlichung des Begriffs atomarer Verhaltensraum zeigt Abbildung 2-2 eine beispielhafte Szenerie mit zwei Fahrstreifen. Aus dieser Szenerie resultieren vier atomare Verhaltensräume. Die longitudinale Unterteilung der Fahrstreifen in jeweils zwei atomare Verhaltensräume resultiert aus der Geschwindigkeitsbegrenzung, die in diesem Beispiel ein Indikationselement repräsentiert.

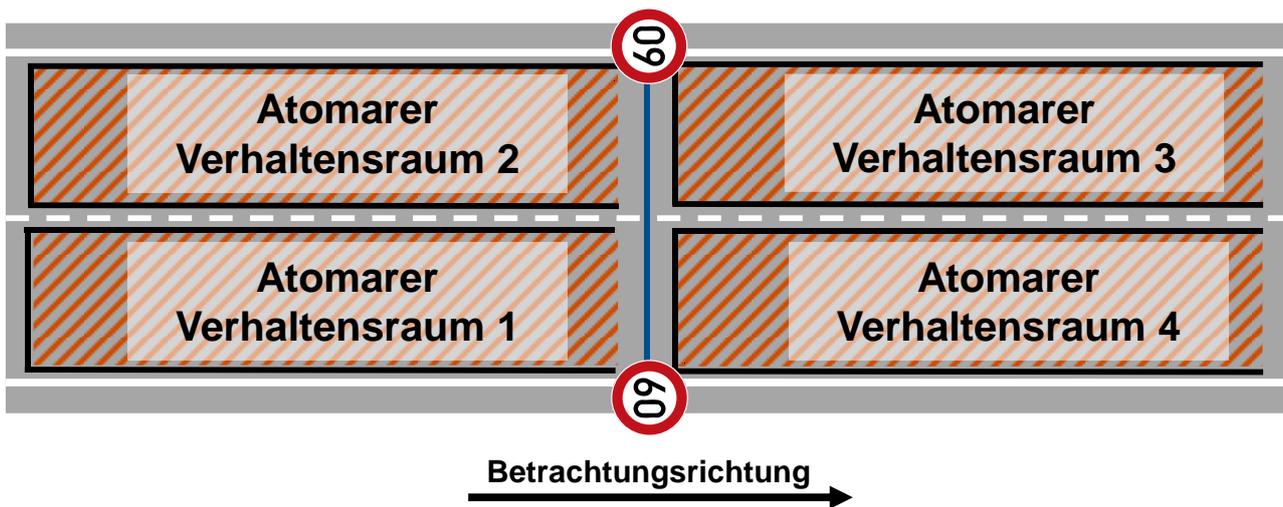


Abbildung 2-2: Szenerie mit vier atomaren Verhaltensräumen

Abbildung 2-3 zeigt die vier Verhaltensattribute der BSSD sowie jeweils dazugehörige Indikationselemente. Die einzelnen Verhaltensattribute werden im Folgenden näher erläutert. Die Beschreibungen in den folgenden Abschnitten 2.3.1 bis 2.3.5 werden ausschließlich der zu der BSSD zugehörigen Veröffentlichung von Glatzki et al.<sup>21c</sup> entnommen.

### 2.3.1 Speed Limit

Das Verhaltensattribut *Speed Limit* legt die zulässige Höchstgeschwindigkeit innerhalb des atomaren Verhaltensraums fest. Dabei ist die zulässige Höchstgeschwindigkeit über weitere Bedingungen, wie beispielsweise eine nasse Straße oder einen bestimmten Zeitraum, veränderbar. Beispielhafte zum Verhaltensattribut *Speed Limit* zugehörige Indikationselemente sind ein Verkehrsschild, Schilder, welche die Art einer Straße (z.B. innerstädtische Straße, Autobahn) angeben, sowie Straßenmarkierungen (siehe linke Spalte in Abbildung 2-3).

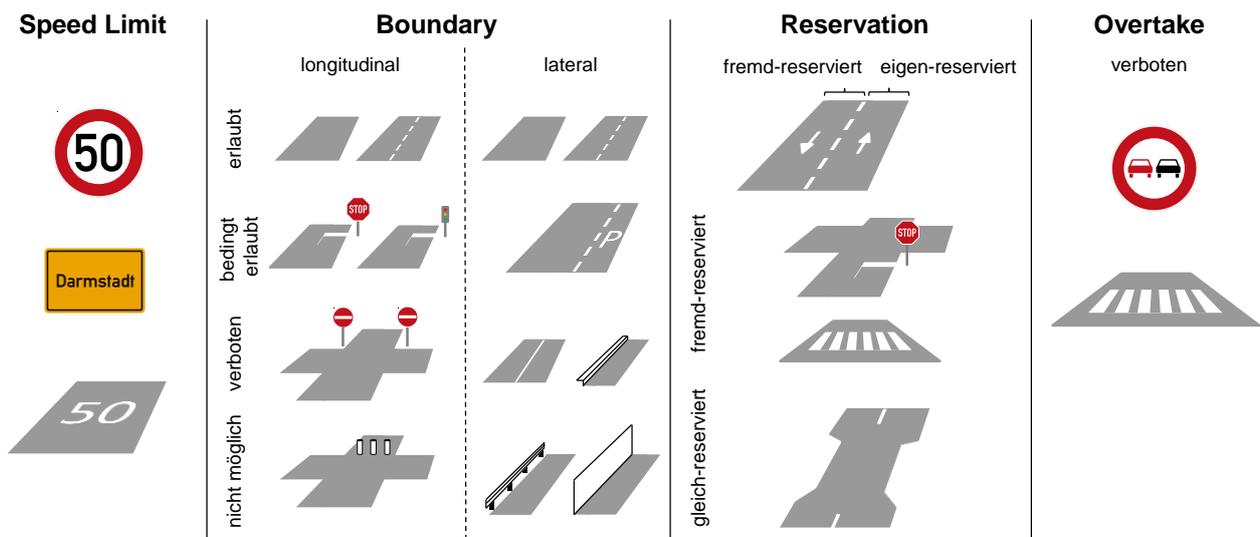


Abbildung 2-3: Verhaltensattribute der BSSD und dazugehörige Indikationselemente<sup>25a</sup>

### 2.3.2 Boundary

Das Verhaltensattribut *Boundary* beschreibt die Verhaltensgrenzen in Bezug auf das Überqueren der Begrenzungen des atomaren Verhaltensraums. Ein atomarer Verhaltensraum besitzt im Allgemeinen zwei laterale Begrenzungen und eine longitudinale Begrenzung (siehe Abbildung 2-2). Das Überqueren einer dieser Begrenzungen ist entweder erlaubt, unter Bedingungen erlaubt, verboten oder nicht möglich.

Die longitudinale Begrenzung beschreibt die Regel bezüglich des Eintritts in den atomaren Verhaltensraum in longitudinaler Richtung. Das Überqueren der longitudinalen Begrenzung ist grundsätzlich immer erlaubt, wenn keine Indikationselemente auftreten und in der zu dem atomaren Verhaltensraum zugehörigen Fahrtrichtung gefahren wird. Beim Auftreten von Indikationselementen besteht die Möglichkeit, dass das Überqueren der longitudinalen Begrenzung unter Bedingungen erlaubt oder verboten ist. Beispielsweise eine Lichtsignalanlage ist ein Indikationselement, welches das Überqueren der longitudinalen Begrenzung unter der Bedingung, dass die Lichtsignalanlage nicht Rot zeigt, erlaubt. Das Überqueren der longitudinalen Begrenzung ist beispielsweise bei einem Verkehrsschild „Verbot der Einfahrt“ verboten. Physische Gegenstände, wie beispielsweise mehrere nebeneinander angeordnete Pfosten, führen zu einem Überqueren der longitudinalen Begrenzung, das nicht möglich ist. Die genannten beispielhaften Indikationselemente sind in Abbildung 2-3 im linken Teil der zweiten Spalte von links dargestellt.

Die lateralen Begrenzungen beschreiben die Regeln bezüglich des Verlassens eines atomaren Verhaltensraums in lateraler Richtung (links und rechts). Die zu den lateralen Begrenzungen zugehörigen Indikationselemente werden durch alle Arten von Fahrstreifenbegrenzungen repräsentiert. Beispielsweise gestrichelte oder nicht vorhandene Fahrstreifenmarkierungen erlauben das Überqueren einer

<sup>25</sup> Mit Änderungen entnommen aus: Glatzki, F. et al.: Behavior-Semantic Scenery Description (BSSD) (2021), a: S. 670; b: S. 671.

---

lateralen Begrenzung eines atomaren Verhaltensraums. Eine durchgezogene Fahrstreifenmarkierung oder ein Bordstein hingegen verbieten das Überqueren der lateralen Begrenzung. Bei einer gestrichelten Fahrstreifenmarkierung, die der Abgrenzung eines Parkplatzes dient, ist das Überqueren einer lateralen Begrenzung unter der Bedingung erlaubt, dass das Ziel besteht, dort zu parken. Das Überqueren einer lateralen Begrenzung ist beispielsweise bei Indikationselementen wie Wänden, Leitplanken oder Gräben nicht möglich. Die genannten beispielhaften Indikationselemente sind in Abbildung 2-3 im rechten Teil der zweiten Spalte von links dargestellt.

### 2.3.3 Reservation

Das Verhaltensattribut *Reservation* legt Bedingungen fest, deren Erfüllung das Eintreten in einen atomaren Verhaltensraum und/oder das Verweilen in diesem aus Vorrangssicht ermöglicht. Jeder atomare Verhaltensraum besitzt eine oder mehrere Klassen von Verkehrsteilnehmern, wie beispielsweise Kraftfahrzeuge, Fahrradfahrer, Fußgänger oder Schienenfahrzeuge, für die der jeweilige atomare Verhaltensraum reserviert ist. Der Begriff *reserviert* meint dabei, dass diese Klasse bzw. diese Klassen von Verkehrsteilnehmern in ihrer Bewegung in dem atomaren Verhaltensraum nicht von anderen Klassen von Verkehrsteilnehmern gestört werden sollten. Ein Verkehrsteilnehmer darf sich nur dauerhaft innerhalb eines atomaren Verhaltensraums aufhalten, wenn er eine Reservierung für diesen atomaren Verhaltensraum besitzt. So ist es einem bestimmten Verkehrsteilnehmer beispielsweise nicht erlaubt, dauerhaft in dem Fahrstreifen der entgegengesetzten Fahrtrichtung zu fahren, da er keine Reservierung für diesen Fahrstreifen bzw. diesen atomaren Verhaltensraum besitzt.

Aus Sicht eines am Straßenverkehr teilnehmenden Ego-Fahrzeugs nehmen atomare Verhaltensräume drei Arten von Reservierungen an. Ein atomarer Verhaltensraum ist *eigen-reserviert*, wenn das Ego-Fahrzeug eine Reservierung für diesen besitzt. Dies ist beispielsweise bei einem Fahrstreifen der Fall, der in der vorgesehenen Fahrtrichtung befahren wird. Besitzt ein anderer Verkehrsteilnehmer beliebiger Klasse eine Reservierung für den atomaren Verhaltensraum, ist dieser *fremd-reserviert*. Ein fremd-reservierter atomarer Verhaltensraum wird beispielsweise durch einen Fahrstreifen repräsentiert, über den ein Fußgängerüberweg führt. In diesem Fall besitzt die Klasse Fußgänger die Reservierung für diesen atomaren Verhaltensraum. Besitzen sowohl das Ego-Fahrzeug als auch weitere Verkehrsteilnehmer eine Reservierung für den betrachteten atomaren Verhaltensraum, ist dieser *gleich-reserviert*. Dies tritt beispielsweise bei zwei sich gegenseitig überlagernden Linksabbiege-Fahrstreifen auf.

### 2.3.4 Overtake

Das Verhaltensattribut *Overtake* definiert die Erlaubnis bezüglich des Überholens von anderen Verkehrsteilnehmern innerhalb eines atomaren Verhaltensraums. Überholen bedeutet in diesem Kontext den Vorgang, bei dem ein Verkehrsteilnehmer an einem anderen sich in die gleiche Richtung fortbewegenden Verkehrsteilnehmer vorbeifährt. Dabei spielt es keine Rolle, ob sich die beiden Verkehrsteilnehmer innerhalb des gleichen Fahrstreifens fortbewegen. Das Überholen eines anderen Verkehrsteilnehmers ist entweder erlaubt oder verboten. Beispielhafte Indikationselemente für das

Verhaltensattribut *Overtake* sind in der rechten Spalte von Abbildung 2-3 dargestellt. Typischerweise handelt es sich dabei um das Verkehrsschild Überholverbot bzw. dessen Aufhebung. Darüber hinaus legt beispielsweise ein Fußgängerüberweg ein Überholverbot fest.

### 2.3.5 Anwendungsbeispiel

Die Verwendung der soeben beschriebenen Verhaltensattribute wird anhand von drei realen Anwendungsbeispielen in Abbildung 2-4 verdeutlicht. Die Abbildung zeigt drei verschiedene Kreuzungen in Darmstadt als Sattelitenaufnahme sowie den aus den dargestellten Kreuzungen resultierenden Verhaltensraum. Kreuzung 1 wird durch eine Lichtsignalanlage gesteuert. In diesem Anwendungsfall wird jedoch davon ausgegangen, dass die Lichtsignalanlage ausgeschaltet ist und demnach das dargestellte Verkehrsschild der Vorfahrtsstraße gültig ist. Kreuzung 2 besteht aus einer Vorfahrtsstraße sowie zwei auf die Vorfahrtsstraße führenden Straßen. Kreuzung 3 zeigt eine unregelmäßige Kreuzung bei der in Deutschland entsprechend der Regelung „rechts vor links“ dem von rechts kommenden Verkehrsteilnehmer Vorfahrt gewährt wird. Dabei ist die nach rechts führende Straße eine Einbahnstraße. Für alle drei Kreuzungen wird ein Linksabbiege-Manöver betrachtet, das durch den orangenen Pfeil symbolisiert wird.

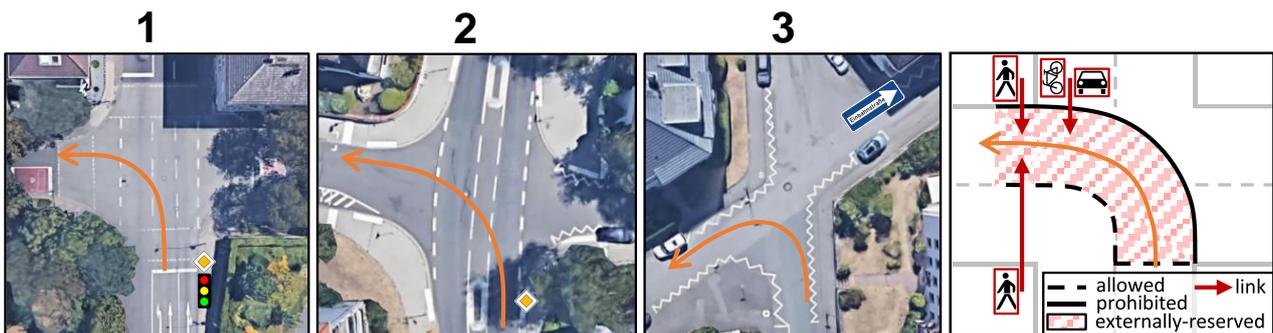


Abbildung 2-4: Anwendungsbeispiel des Verhaltensraums der BSSD<sup>25b</sup>

In allen drei Kreuzungen gilt entsprechend des zugehörigen Verkehrsschildes (nicht in Abbildung 2-4 dargestellt) eine Geschwindigkeitsbegrenzung von 30 km/h. Aufgrund dessen wird das Verhaltensattribut *Speed Limit* mit dem Wert 30 km/h belegt. Weiterhin besteht in allen drei Kreuzungen kein Überholverbot. Daher wird das Verhaltensattribut *Overtake* mit dem Wert *erlaubt* belegt.

Die Kreuzungen 2 und 3 dürfen generell aus der mit dem Pfeil dargestellten Richtung befahren werden. Für Kreuzung 1 gilt dies ebenfalls, da die dargestellte Lichtsignalanlage ausgeschaltet ist. Aufgrund dessen wird bei allen drei Kreuzungen der longitudinalen Begrenzung des Verhaltensattributs *Boundary* der Wert *erlaubt* zugewiesen. Für das Durchführen des gewünschten Linksabbiege-Manövers ist das Überqueren der linken lateralen Begrenzungen des vom Fahrzeug befahrenen Fahrstreifens erlaubt. Während des Linksabbiege-Manövers ist das Überqueren der rechten lateralen Begrenzung verboten. Demnach wird die linke laterale Begrenzung des Verhaltensattributs *Boundary* auf den Wert *erlaubt* gesetzt sowie die rechte laterale Begrenzung auf den Wert *verboten* gesetzt.

Die Belegung des Verhaltensattributs *Reservation* wird durch die in den dargestellten Kreuzungen geltenden Verkehrsregeln bestimmt. In den Kreuzungen 1 und 2 besitzen Verkehrsteilnehmer, die das

---

Linksabbiege-Manöver durchführen, aufgrund des Verkehrsschildes der Vorfahrtsstraße Vorrang gegenüber Verkehrsteilnehmern, die von der linken und rechten auf die Kreuzung auftreffenden Straße kommen. Weiterhin müssen die das Linksabbiege-Manöver durchführenden Verkehrsteilnehmer anderen aus der entgegengesetzten Richtung der Vorfahrtsstraße kommenden Verkehrsteilnehmern sowie Fußgängern, welche die Zielstraße des Linksabbiege-Manövers parallel zur Vorfahrtsstraße überqueren, Vorrang gewähren. In Kreuzung 3 ist ebenfalls kein Gewähren von Vorrang für die von links und die von rechts kommenden Verkehrsteilnehmer notwendig („rechts vor links“ bzw. Einbahnstraße). Wie bei den Kreuzungen 1 und 2 muss bei Durchführung des Linksabbiege-Manövers allen entgegenkommenden Verkehrsteilnehmern sowie Fußgängern, welche die Zielstraße des Linksabbiege-Manövers parallel zur Richtung der Einfahrtsstraße des betrachteten Manövers überqueren, Vorrang gewährt werden. In allen drei Kreuzungen ist Platz für die parallele Durchführung von zwei Linksabbiege-Manövern aus entgegengesetzten Richtungen vorhanden. Zusammenfassend ergibt sich die Belegung des Verhaltensattributs *Reservierung* mit dem Wert *fremd-reserviert*. Dabei werden die Klassen Fußgänger, Fahrradfahrer und Kraftfahrzeug unter Angabe ihrer Herkunft und ihres Ziels als die die Reservierung besitzenden Verkehrsteilnehmer angegeben.

Der aus den drei beschriebenen Linksabbiege-Manövern resultierende Verhaltensraum ist rechts in Abbildung 2-4 visualisiert. Es ist zu erkennen, dass alle drei Manöver den identischen Verhaltensraum besitzen, sofern die Lichtsignalanlage in Kreuzung 1 ausgeschaltet ist. Somit bietet die BSSD die Möglichkeit, unterschiedliche Szenerien, welche die gleichen Verhaltensgrenzen für ein Fahrzeug festlegen, über eine einheitliche Beschreibung zu repräsentieren und somit die Komplexität der gesamten Beschreibung der Szenerie zu reduzieren.

## 2.4 Definitionsgrenzen

Für die Integration der BSSD in ein hochgenaues Kartenformat ist erforderlich festzulegen, für welche Teile eines Verkehrsnetzwerks die BSSD modelliert bzw. nicht modelliert wird. Da die BSSD das Ziel verfolgt, die Verhaltensgrenzen für ein in einer bestimmten Szenerie befindliches Kraftfahrzeug zu definieren, ist es nur möglich, die BSSD für die Teile des Verkehrsnetzwerks zu modellieren, die mit einem Kraftfahrzeug zu befahren sind. Für eine eindeutige Definition bezüglich der mit einem Kraftfahrzeug zu befahrenden bzw. nicht zu befahrenden Bereiche eines Verkehrsnetzwerks wird im Rahmen dieser Arbeit festgelegt, dass die BSSD ausschließlich für die *Fahrbahn* eines Verkehrsnetzwerks modelliert wird. Dazu wird die von der Forschungsgesellschaft für Straßen- und Verkehrswesen (FGSV) festgelegte Definition des Begriffs Fahrbahn verwendet, die eine Fahrbahn als den „aus Fahrstreifen und Randstreifen bestehende[n], zusammenhängend[en] befestigte[n] Teil der Straße“<sup>26a</sup> bezeichnet. Dabei bezeichnet ein Fahrstreifen den „durch Markierung gekennzeichnete[n] Teil der Fahrbahn, der für die Fortbewegung einer Fahrzeugreihe bestimmt ist“<sup>26a</sup>. Ein Randstreifen bezeichnet den „unmittelbar neben den Fahrstreifen liegende[n], optisch gekennzeichnete[n] Teil der Fahrbahn, der die Fahrbahn seitlich begrenzt“<sup>26b</sup>. Zur Verdeutlichung dieser drei Begriffe ist in Abbildung

---

<sup>26</sup> FGSV: Begriffsbestimmungen für das Straßen- und Verkehrswesen (2020), a: S.77; b: S.80.

2-5 beispielhaft ein Straßenquerschnitt einer Fahrbahn, die aus zwei Fahr- und Randstreifen besteht, dargestellt.

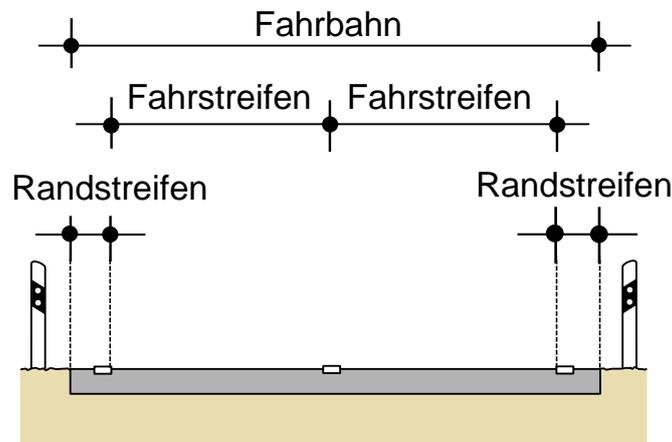


Abbildung 2-5: Straßenquerschnitt einer Fahrbahn<sup>27</sup>

## 2.5 Spezifikation

Für die Speicherung und den Austausch der BSSD wird bei FZD aktuell eine Spezifikation eines auf der Extensible Markup Language (XML) basierenden Dateiformats entwickelt. Im Rahmen dieser Arbeit wird die Spezifikation der BSSD in der Version 0.4<sup>28</sup> verwendet. Die Spezifikation ist eine mögliche Umsetzung der BSSD in topologischer Form. Die Topologie ermöglicht die Beschreibung der Struktur eines Straßennetzes, ohne dabei eine Geometrie zu definieren. In diese Struktur wird der eigentliche Verhaltensraum der BSSD eingefügt. In einer topologischen Beschreibung wird ein Straßennetz durch Kanten und Knoten abstrakt repräsentiert. Ein Knoten verbindet dabei jeweils mindestens drei Kanten. Somit ist eine anschauliche Zuordnung der Elemente Kante und Knoten zu den Elementen Straße und Kreuzung eines realen Straßennetzes möglich. Ein beispielhaftes Straßennetz, das durch Knoten und Kanten repräsentiert wird, ist in Abbildung 2-6 dargestellt. Die Knoten werden durch die grauen Punkte dargestellt. Die Kanten werden durch die blauen Linien zwischen den Punkten repräsentiert.

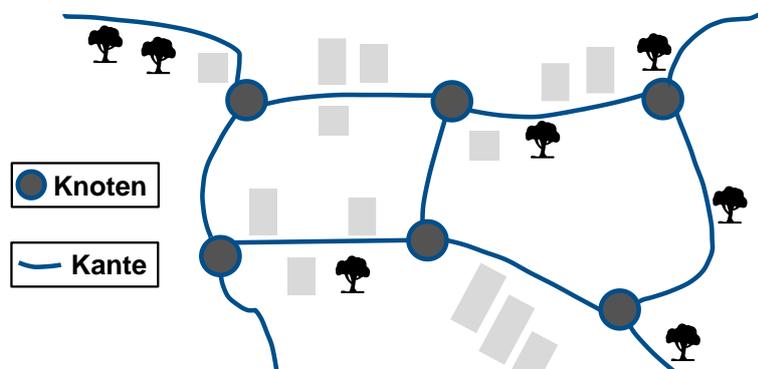


Abbildung 2-6: Topologische Repräsentation eines Straßennetzes

<sup>27</sup> Mit Änderungen entnommen aus: Pöschl, S.: Straßenquerschnitt (2008).

<sup>28</sup> Glatzki, F.; Lippert, M.: Behavior-Semantic Scenery Description v0.4 (unveröffentlicht) (2021).

Wie bereits erwähnt, basiert die Spezifikation der BSSD auf dem Dateiformat XML. Somit besteht eine BSSD-Datei aus mehreren hierarchisch strukturierten Elementen und einzelnen, den Elementen zugeordneten Attributen. Die hierarchische Struktur der BSSD-Spezifikation ist in Abbildung 2-7 dargestellt. Es existieren die drei übergeordneten Elemente `<header>`, `<way>` und `<node>`. Im Element `<header>` werden allgemeine Informationen zu der Datei, wie beispielsweise der Dateiname und das Erstellungsdatum, gespeichert. Das `<way>`-Element repräsentiert eine Kante bzw. einen Weg<sup>29</sup> des topologischen Straßennetzes. Das `<node>`-Element repräsentiert einen Knoten des topologischen Straßennetzes.<sup>30</sup> Die den Verhaltensattributen der BSSD zugehörigen Elemente sind in Orange dargestellt. Die jeweils über einem Element dargestellte Zahl bzw. der jeweils über einem Element dargestellte Zahlenbereich gibt an, wie viele Entitäten dieses Elements existieren müssen bzw. dürfen. Dabei repräsentiert das Symbol \* eine beliebige Anzahl.

Im Folgenden wird die BSSD-Spezifikation näher beschrieben. Dazu werden die Elemente `<way>` und `<node>` inklusive der zu diesen Elementen untergeordneten Elementen sowie deren Attribute betrachtet. Die den Verhaltensattributen der BSSD entsprechenden Elemente werden in einem separaten Abschnitt erläutert. Alle Informationen in den folgenden Abschnitten wurden der Spezifikation der BSSD in der Version 0.4<sup>28</sup> entnommen.

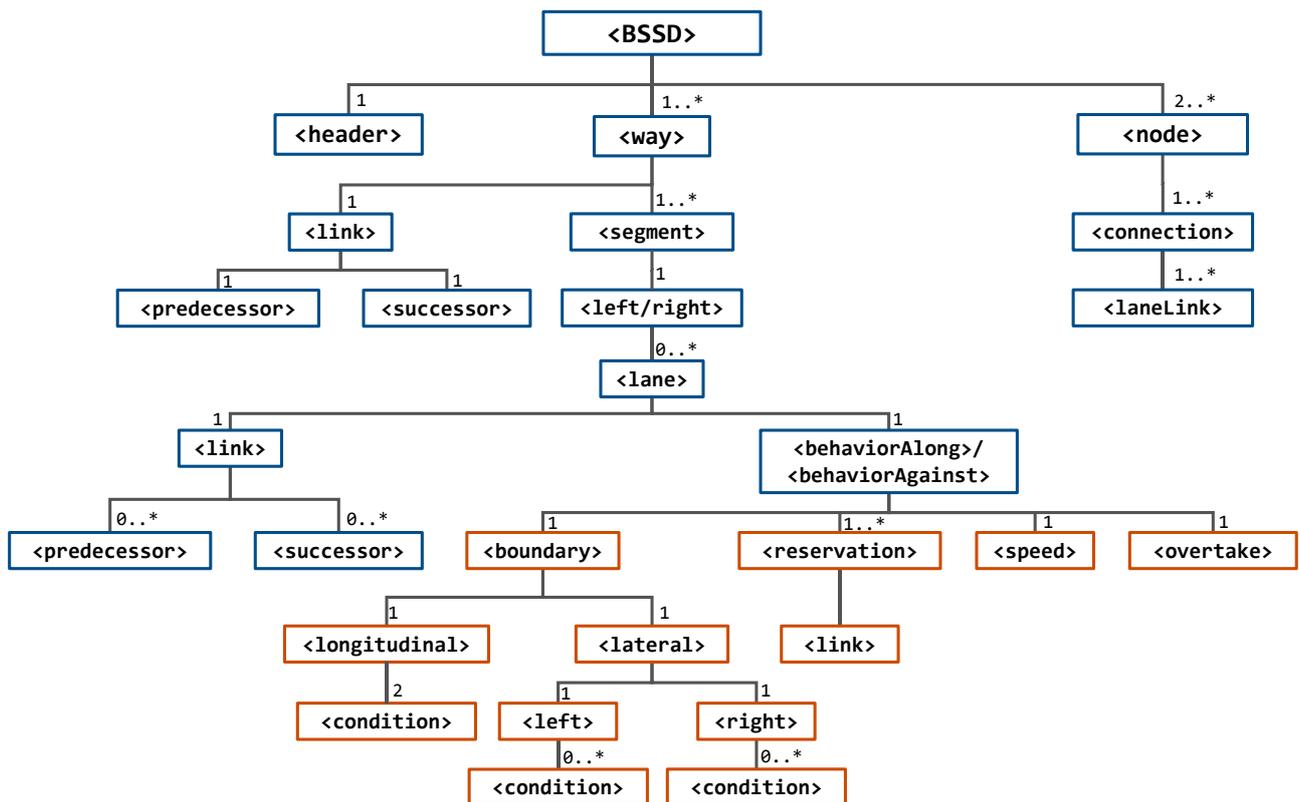


Abbildung 2-7: Hierarchische Struktur der BSSD-Spezifikation<sup>31</sup>

<sup>29</sup> Im Rahmen der BSSD-Spezifikation wird eine Kante mit dem Begriff *Weg* bezeichnet

<sup>30</sup> In diesem Kapitel werden die Begriffe *Weg* und *<way>-Element* bzw. *Knoten* und *<node>-Element* äquivalent verwendet

<sup>31</sup> Eigene Darstellung nach: Glatzki, F.; Lippert, M.: Behavior-Semantic Scenery Description v0.4 (unveröffentlicht) (2021), S. 5.

---

## 2.5.1 Das <way>-Element

Ein Weg des topologischen Straßennetzes wird über das **<way>**-Element repräsentiert. Das zu einer BSSD-Datei zugehörige topologische Straßennetz besitzt immer mindestens einen Weg. Zur Charakterisierung eines Wegs besitzt das **<way>**-Element vier verpflichtende Attribute: Das Attribut **id** beinhaltet eine eindeutige Nummer zur Identifizierung des Wegs innerhalb der BSSD-Datei. Die Attribute **from** und **to** geben die **id**-Attribute der beiden Knoten an, die durch den Weg miteinander verbunden werden. Gleichzeitig definieren diese beiden Attribute eine für diesen Weg eindeutige Referenzrichtung, die für die Definition der Verhaltensattribute relevant ist. Die Referenzrichtung führt von dem im Attribut **from** referenzierten Knoten in Richtung des im Attribut **to** referenzierten Knotens. Das Attribut **node** gibt die **id** des Knotens an, der den Weg beinhaltet, falls der Weg innerhalb eines Knotens definiert ist (siehe Abschnitt 2.5.2.3). Falls der Weg außerhalb eines Knotens definiert ist, nimmt das Attribut **node** den Wert "-1" an.

### 2.5.1.1 Das <link>-Element

Jeder Weg hat genau ein untergeordnetes **<link>**-Element, das den Weg mit anderen Wegen und Knoten verknüpft und somit den Aufbau des topologischen Straßennetzes ermöglicht. Dazu besitzt das **<link>**-Element jeweils ein untergeordnetes Element **<predecessor>** und **<successor>**, welches den vorangehenden bzw. nachfolgenden Weg oder Knoten<sup>32</sup> angibt. Hierfür besitzt ein **<predecessor>**- bzw. **<successor>**-Element die verpflichtenden Attribute **id** und **element-*Type***, welche die **id** des Vorgänger- bzw. Nachfolgerelements und den Typ des Vorgänger- bzw. Nachfolgerelements (Weg oder Knoten) angeben. Ein Weg, der sich außerhalb eines Knotens befindet, besitzt immer jeweils einen Knoten als Vorgänger- und Nachfolgerelement. Ein Weg, der sich innerhalb eines Knotens befindet (siehe Abschnitt 2.5.2.3), besitzt immer jeweils einen Weg als Vorgänger- und Nachfolgerelement.

### 2.5.1.2 Das <segment>-Element

Ein Weg besitzt mindestens ein untergeordnetes **<segment>**-Element. Ein Segment<sup>33</sup> repräsentiert einen in longitudinaler Richtung abgegrenzten Teil eines Wegs. Es beinhaltet einen oder mehrere lateral nebeneinander angeordnete atomare Verhaltensräume. Somit bilden der Anfang und das Ende des Segments die longitudinalen Begrenzungen der in dem Segment enthaltenen atomaren Verhaltensräume (siehe Abbildung 2-8). Innerhalb des Segments ist der Verhaltensraum in longitudinaler Richtung konstant. Sobald in longitudinaler Richtung eine Änderung eines Verhaltensattributs auftritt, wird ein neues Segment definiert. Zur eindeutigen Identifikation eines Segments besitzt dieses das verpflichtende Attribut **id**, das einem Segment eine innerhalb des Wegs eindeutige Nummer zuweist. Einem Segment sind immer ein Element **<left>** und ein Element **<right>** untergeordnet, die keine Attribute besitzen und wiederum mehrere untergeordnete **<lane>**-Elemente besitzen

---

<sup>32</sup> Die Festlegung von Vorgänger- und Nachfolgerelement eines Wegs ergibt sich durch Betrachtung von dessen Referenzrichtung

<sup>33</sup> In diesem Kapitel werden die Begriffe *<segment>-Element* und *Segment* äquivalent verwendet

---

können. Die Elemente `<left>` und `<right>` gruppieren die untergeordneten `<lane>`-Elemente in linke und rechte `<lane>`-Elemente (aus Sicht der Mittellinie des Wegs in Referenzrichtung), um die Navigation eines menschlichen Lesers durch die BSSD-Datei zu vereinfachen.

Ein `<lane>`-Element ist ein in lateraler Richtung abgegrenzter Teilbereich eines Segments. Somit repräsentiert es eine Struktur, die einen atomaren Verhaltensraum beinhaltet. Der in dem `<lane>`-Element enthaltene atomare Verhaltensraum ist in longitudinaler Richtung durch den Anfang und das Ende des Segments und in lateraler Richtung durch die linke und rechte Begrenzung des `<lane>`-Elements begrenzt. Wie in Abschnitt 2.3 bereits erwähnt, entspricht ein atomarer Verhaltensraum meist einem in longitudinaler Richtung abgegrenzten Teils eines Fahrstreifens. Daraus erklärt sich die Bezeichnung dieser Struktur als *<Lane>-Element*<sup>34</sup>.

Zur Verdeutlichung der Strukturen Weg, Segment und Fahrstreifen ist in Abbildung 2-8 eine beispielhafte Straße dargestellt, die einen Weg eines topologischen Straßennetzes repräsentiert. Hierbei sei erneut darauf hingewiesen, dass die dargestellten geometrischen Dimensionen der Straße sowie der darin enthaltenen Fahrstreifen nicht durch die Spezifikation der BSSD definiert werden. Der dargestellte Weg ist in longitudinaler Richtung in zwei Segmente aufgeteilt, die jeweils vier lateral nebeneinander angeordnete Fahrstreifen beinhalten. Die Fahrstreifen werden in longitudinaler Richtung durch die Segmente und in lateraler Richtung durch eine linke und rechte Fahrstreifenmarkierung begrenzt. Die Segmentierung des Wegs ergibt sich dadurch, dass die gestrichelte Fahrstreifenmarkierung zwischen den Fahrstreifen mit der `id -1` und `-2` in eine durchgezogene Fahrstreifenmarkierung wechselt. Entsprechend der StVO werden dadurch die Verhaltensattribute *Overtake* und *Boundary* verändert und somit ist die Definition eines neuen Segments notwendig. Dieses Beispiel zeigt darüber hinaus, dass eine einheitliche Segmentierung für beide Fahrrichtungen durchgeführt wird. Obwohl sich nur bei den zu einer Fahrrichtung zugehörigen Fahrstreifen mit der `id -1` und `-2` die Verhaltensattribute ändern, werden die Fahrstreifen der entgegengesetzten Fahrrichtung (`id 1` und `2`) ebenfalls durch die Segmentierung aufgetrennt. Folglich ist der atomare Verhaltensraum in den Fahrstreifen mit der `id 1` und `2` jeweils identisch in Segment A und Segment B.

Jeder Fahrstreifen verfügt über das verpflichtende Attribut `id`, das eine eindeutige Identifikation des Fahrstreifens in dem betrachteten Segment ermöglicht. Darüber hinaus ist anhand dieses Attributs die Bestimmung der lateralen Anordnung der Fahrstreifen innerhalb des Segments sowie die Bestimmung der dem Fahrstreifen zugehörigen Fahrrichtung möglich. Fahrstreifen, deren Fahrrichtung der Referenzrichtung entspricht, besitzen beim Gelten von Rechtsverkehr eine `id` mit einem negativen Wert. Der Startwert des innersten Fahrstreifens ist `-1`. Die `id`-Attribute der nach Außen in lateraler Richtung folgenden Fahrstreifen werden mit absteigender Nummerierung belegt (siehe Abbildung 2-8). Das Vorgehen für Fahrstreifen, deren Fahrrichtung der entgegengesetzten Referenzrichtung entspricht, ist analog mit positiven `id`-Attributen und aufsteigender Nummerierung. Falls ein Fahrstreifen zwei zugehörige Fahrrichtungen besitzt, wird das Attribut `id` mit dem Wert `-1` belegt.

---

<sup>34</sup> In diesem Kapitel werden die Begriffe *<Lane>-Element* und *Fahrstreifen* äquivalent verwendet

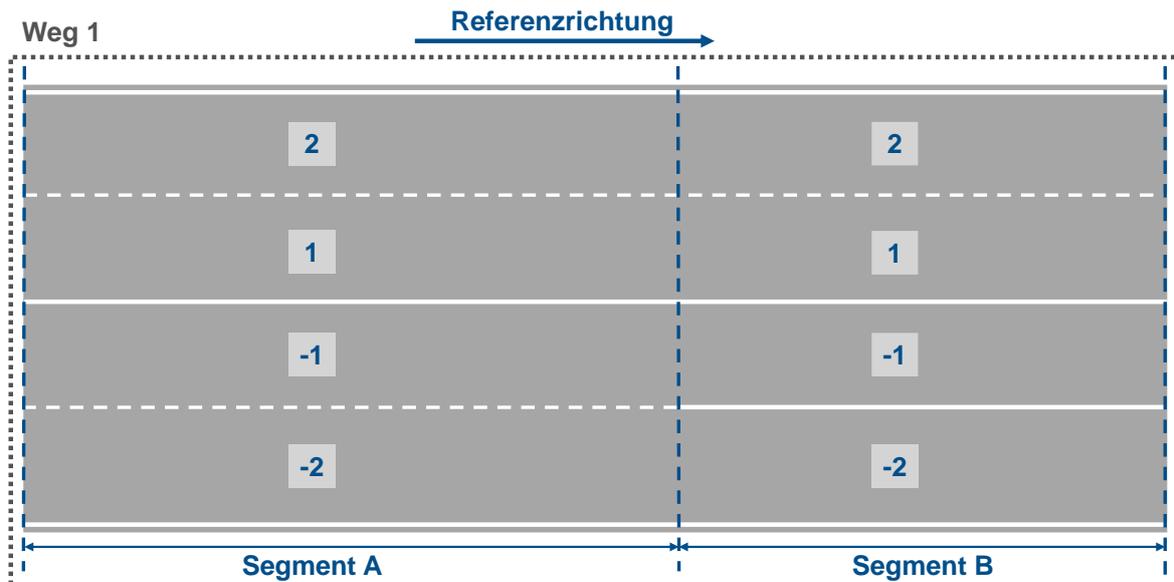


Abbildung 2-8: Aufteilung eines Wegs in Segmente und Fahrstreifen bei der BSSD

Analog zum `<way>`-Element verfügt ein `<lane>`-Element über ein untergeordnetes `<link>`-Element, das den Fahrstreifen mit in longitudinaler Richtung vorangehenden und nachfolgenden Fahrstreifen<sup>35</sup> verknüpft. Die Angabe der vorangehenden und nachfolgenden Fahrstreifen erfolgt über die Elemente `<predecessor>` und `<successor>`, die jeweils das Attribut `laneId` besitzen, das die `id` des vorangehenden bzw. nachfolgenden Fahrstreifens angibt. Das `<link>`-Element besitzt eine beliebige Anzahl an untergeordneten `<predecessor>`- und `<successor>`-Elementen. Ein Fahrstreifen besitzt beispielweise kein Element `<predecessor>` oder `<successor>`, wenn dieser Fahrstreifen aus einem Knoten entspringt bzw. in einem Knoten endet. Ein Beispiel für einen Fahrstreifen mit einem vorangehenden bzw. einem nachfolgenden Fahrstreifen sind alle in Abbildung 2-8 dargestellten Fahrstreifen. Wenn sich ein Fahrstreifen in zwei Fahrstreifen aufteilt, werden mehrere vorangehende bzw. nachfolgende Fahrstreifen angegeben.

## 2.5.2 Die Elemente des Verhaltensraums

Ein `<lane>`-Element besitzt genau ein untergeordnetes Element `<behaviorAlong>` und ein untergeordnetes Element `<behaviorAgainst>`. Diese beiden Elemente enthalten jeweils die vier den Verhaltensattributen der BSSD zugehörigen Elemente `<boundary>` (Verhaltensattribut *Boundary*), `<reservation>` (Verhaltensattribut *Reservation*), `<speed>` (Verhaltensattribut *Speed Limit*) und `<overtake>` (Verhaltensattribut *Overtake*). Im Element `<behaviorAlong>` werden die Verhaltensattribute aus Sicht eines Fahrzeugs, das in Referenzrichtung fährt, für den im `<lane>`-Element enthaltenen atomaren Verhaltensraum definiert. Entsprechend werden im Element `<behaviorAgainst>` die Verhaltensattribute aus Sicht eines entgegen der Referenzrichtung fahrenden Fahrzeugs definiert. Die vier den Verhaltensattributen zugehörigen Elemente werden im Folgenden näher erläutert.

<sup>35</sup> Die Festlegung von vorangehenden und nachfolgenden Fahrstreifen ergibt sich durch Betrachtung der Referenzrichtung des Wegs

---

### 2.5.2.1 Boundary

Das Verhaltensattribut *Boundary* wird durch das **<boundary>**-Element repräsentiert. Die Unterscheidung in die Verhaltensvorgabe bezüglich des Überquerens der longitudinalen und der lateralen Begrenzungen des atomaren Verhaltensraums, erfolgt durch die dem **<boundary>**-Element untergeordneten Elemente **<longitudinal>** und **<lateral>**.

Das **<longitudinal>**-Element verfügt über maximal drei Attribute: Das verpflichtende Attribut **crossing** definiert die Regel bezüglich des Überquerens der longitudinalen Begrenzung zum Eintritt in den atomaren Verhaltensraum. Wie bereits in Abschnitt 2.3.2 erwähnt, ist das Überqueren der longitudinalen Begrenzung erlaubt, unter Bedingungen erlaubt, verboten oder nicht möglich, woraus die möglichen Werte "allowed", "condition", "prohibited" oder "not possible" des Attributs **crossing** resultieren. Wird der longitudinale Eintritt in den atomaren Verhaltensraum über eine Lichtsignalanlage geregelt, wird das optionale Attribut **trafficLightActive** definiert. In diesem Fall werden zwei **<longitudinal>**-Elemente definiert. In einem der beiden **<longitudinal>**-Elemente wird der Fall betrachtet, dass die Lichtsignalanlage aktiv ist (**trafficLightActive="yes"**). Im anderen **<longitudinal>**-Element wird folglich der Fall betrachtet, dass die Lichtsignalanlage ausgeschaltet ist (**trafficLightActive="no"**). Für den Fall, dass der longitudinale Eintritt über eine Lichtsignalanlage geregelt wird und der Eintritt bei roter Lichtsignalanlage erlaubt ist (Verkehrszeichen 720 StVO, *Grünpfeil*), wird das optionale Attribut **redLightCondition** mit dem Wert "yes" definiert.

Ist das Überqueren der longitudinalen Begrenzung unter Bedingungen erlaubt, besitzt das **<longitudinal>**-Element das untergeordnete Element **<condition>**. Ein **<condition>**-Element definiert die Bedingung unter der das Überqueren der longitudinalen Begrenzung erlaubt ist. Dazu besitzt es das verpflichtende Attribut **type**, das einen beliebigen Freitext enthält, der die Bedingung beschreibt. Mögliche Bedingungen sind beispielsweise, dass das Fahrzeug angehalten hat (**type="stop"**) oder dass die Lichtsignalanlage kein Rot zeigt (**type="no red light"**). Es ist möglich, ein oder zwei **<condition>**-Elemente zu definieren. Falls zwei **<condition>**-Elemente definiert sind, ist das gleichzeitige Erfüllen beider Bedingungen notwendig, damit der longitudinale Eintritt erlaubt ist. Die Attribute des **<longitudinal>**-Elements sind zusammenfassend in Tabelle 2-1 gelistet.

Das **<lateral>**-Element verfügt über die beiden untergeordneten Elemente **<left>** und **<right>**. Diese definieren die Regelung bezüglich des Verlassens des atomaren Verhaltensraums über die linke bzw. rechte<sup>36</sup> laterale Begrenzung des atomaren Verhaltensraums. Analog zum Element **<longitudinal>** verfügen die Elemente **<left>** und **<right>** jeweils über das Attribut **crossing**, das einen der Werte "allowed", "condition", "prohibited" oder "not possible" annimmt. Daraus folgend besitzen auch die Elemente **<left>** und **<right>** jeweils das untergeordnete Element **<condition>** mit dem dazugehörigen Attribut **type**, falls der Austritt über die linke bzw. rechte

---

<sup>36</sup> Die Festlegung der linken und rechten lateralen Begrenzung des atomaren Verhaltensraums ergibt sich durch die Betrachtung der durch das Element **<behaviorAlong>** bzw. **<behaviorAgainst>** festgelegten Fahrtrichtung

---

Begrenzung unter Bedingungen erlaubt ist. Im Unterschied zum Element `<longitudinal>` ist es möglich, dem `<left>`- bzw. `<right>`-Element beliebig viele `<condition>`-Elemente unterzuordnen, deren Erfüllung für das Überqueren der lateralen Begrenzung notwendig ist.

### 2.5.2.2 Reservation

Das Verhaltensattribut *Reservation* wird durch das `<reservation>`-Element repräsentiert. Jedes `<reservation>`-Element repräsentiert genau eine Reservierung<sup>37</sup>. Ist der atomare Verhaltensraum eigen-reserviert, existiert folglich nur ein `<reservation>`-Element. Wenn der atomare Verhaltensraum hingegen fremd- oder gleich-reserviert ist, wird für jede Klasse von Verkehrsteilnehmern, die eine Reservierung für diesen atomaren Verhaltensraum besitzen, ein eigenes `<reservation>`-Element definiert.

Ein `<reservation>`-Element besitzt maximal sechs, jedoch mindestens zwei Attribute. Das verpflichtende Attribut `id` beinhaltet eine innerhalb des Elements `<behaviorAlong>` bzw. `<behaviorAgainst>` eindeutige Nummer zur Identifizierung der Reservierung. Die Art der Reservierung wird über das verpflichtende Attribut `type` festgelegt, das entsprechend der Art der Reservierung einen der Werte "own-reserved", "externally-reserved" oder "equally-reserved" annimmt. Falls eine Reservierung extern- oder gleich-reserviert ist, wird das Attribut `object` definiert, das die Klasse des Verkehrsteilnehmers angibt, der die Reservierung in Anspruch nimmt. Dazu nimmt es einen der Werte "pedestrian", "bicycle", "motor vehicle" oder "railed vehicle" entsprechend der Klassen Fußgänger, Fahrradfahrer, Kraftfahrzeug und Schienenfahrzeug an. Analog zum `<longitudinal>`-Element des Verhaltensattributs *Boundary* besitzt das `<reservation>`-Element die beiden optionalen Attribute `trafficLightActive` und `redLightCondition`. Diese werden in diesem Fall definiert, wenn eine Lichtsignalanlage die Reservierung beeinflusst bzw. ein erlaubter Übertritt bei roter Lichtsignalanlage (Verkehrszeichen 720 StVO, *Grünpfeil*) die Reservierung beeinflusst. Zusätzlich zu diesen beiden Sonderfällen verfügt das `<reservation>`-Element über das Attribut `turnArrowActive`, das definiert wird, wenn die Reservierung durch eine Lichtsignalanlage beeinflusst wird und ein zusätzliches Linksabbiegesignal<sup>38</sup> existiert. Entsprechend eines aktiven bzw. ausgeschalteten Linksabbiegesignals nimmt dieses Attribut den Wert "yes" bzw. "no" an. Die Attribute des `<reservation>`-Elements sind zusammenfassend in Tabelle 2-2 gelistet.

Wenn eine Reservierung fremd- oder gleich-reserviert ist, besitzt das `<reservation>`-Element mindestens ein untergeordnetes `<link>`-Element. Dieses Element gibt verpflichtend die Herkunft und optional das Ziel des die Reservierung in Anspruch nehmenden Verkehrsteilnehmers an. Dazu besitzt das `<link>`-Element das verpflichtende Attribut `id`, das eine eindeutige Identifizierung des `<link>`-Elements innerhalb der Reservierung ermöglicht. Die Herkunft des Verkehrsteilnehmers wird über die verpflichtenden Attribute `linkedWay`, `linkedSegment` und `linkedArea` angegeben, welche die `id`-Attribute von Weg, Segment und Fahrstreifen des Herkunfts-Fahrstreifens. Das Ziel des

---

<sup>37</sup> In diesem Abschnitt werden die Begriffe `<reservation>`-Element und *Reservierung* äquivalent verwendet

<sup>38</sup> Das zusätzliche Linkabbiegesignal ermöglicht wartenden Linksabbiegern ein Abfließen aus dem Kreuzungsbereich bei roter Lichtsignalanlage für den Gegenverkehr

Verkehrsteilnehmers wird analog über die optionalen Attribute `targetWay`, `targetSegment` und `targetArea` angegeben. Falls ein Verkehrsteilnehmer mehr als eine mögliche Herkunft bzw. mehr als ein mögliches Ziel besitzt, werden mehrere `<link>`-Elemente für eine Reservierung definiert.

Tabelle 2-1: Attribute `<longitudinal>`-Element

Attribut	Werte	Optional?
<code>crossing</code>	"allowed"; "condition"; "prohibited"; "not possible"	Nein
<code>traffic-Light-Active</code>	"yes"; "no"	Ja
<code>redLight-Condition</code>	"yes"	Ja

Tabelle 2-2: Attribute `<reservation>`-Element

Attribut	Werte	Optional?
<code>id</code>	[1, ∞[	Nein
<code>type</code>	"own-reserved"; "externally-reserved"; "equally-reserved"	Nein
<code>object</code>	"pedestrian"; "bicycle"; "motor vehicle"; "railed vehicle"	Ja
<code>traffic-LightActive</code>	"yes"; "no"	Ja
<code>redLight-Condition</code>	"yes"	Ja
<code>turn-ArrowActive</code>	"yes"; "no"	Ja

### 2.5.2.3 Speed Limit

Das Verhaltensattribut *Speed Limit* wird durch das `<speed>`-Element repräsentiert. Die Angabe der für den atomaren Verhaltensraum geltenden zulässigen Höchstgeschwindigkeit in km/h erfolgt durch die Angabe des verpflichtenden Attributs `max`. Ist zusätzlich eine zulässige Höchstgeschwindigkeit für ein bestimmtes Zeitintervall definiert, existieren die Attribute `timeInterval` und `timeMax`, die das Zeitintervall und die während dieses Zeitintervalls zulässige Höchstgeschwindigkeit angeben. Falls eine zusätzliche zulässige Höchstgeschwindigkeit bei Nässe definiert ist, wird diese im optionalen Attribut `wetMax` angegeben.

### 2.5.2.4 Overtake

Das Verhaltensattribut *Overtake* wird durch das `<overtake>`-Element repräsentiert. Dieses verfügt über das verpflichtende Attribut `permission`, das entsprechend der Erlaubnis bzw. des Verbots zu Überholen den Wert "yes" bzw. "no" annimmt.

## 2.5.3 Das `<node>`-Element

Ein Knoten des topologischen Straßennetzes wird über das `<node>`-Element repräsentiert. Damit der für die Definition eines topologischen Straßennetzes mindestens erforderliche Weg einen eindeutigen Start- und Endpunkt besitzt, sind mindestens zwei Knoten notwendig. Ein Knoten besitzt zwei verpflichtende Attribute: Das Attribut `id` beinhaltet eine Nummer zur eindeutigen Identifikation des

Knotens innerhalb des topologischen Straßennetzes. Das Attribut `infinitesimal` mit den möglichen Ausprägungen "yes" oder "no" dient der Angabe des Sonderfalls eines infinitesimalen Knotens. Ein infinitesimaler Knoten wird verwendet, wenn sich ein Weg ohne eine Knotenfläche in weitere Wege aufteilt (z.B. Autobahnausfahrt, siehe auch Abbildung 3-11).

Wie zuvor erwähnt, verbindet ein Knoten mindestens drei Wege. Dazu besteht der Knoten wiederum aus einzelnen Wegen, im Folgenden als *verbindende Wege* bezeichnet, die jeweils eine definierte Verbindung zwischen zwei Wegen ermöglichen. Die verbindenden Wege werden als reguläre `<way>`-Elemente modelliert (siehe Abschnitt 2.5.1). Im Unterschied zu außerhalb eines Knotens befindlichen Wegen wird im Attribut `node` eines verbindenden Wegs die `id` des zugehörigen Knotens (statt des Werts "-1") angegeben. Die Zusammensetzung eines Knotens aus Wegen ist in Abbildung 2-9 dargestellt, die ein beispielhaftes topologisches Netz bestehend aus vier Knoten und drei Wegen zeigt. Die Pfeilrichtung eines Wegs repräsentiert dessen Referenzrichtung. In dem Beispiel verbindet Knoten 4 die Wege 1, 2 und 3 anhand der verbindenden Wege 12, 13 und 32. Jeder der verbindenden Wege ermöglicht eine definierte Verbindung zwischen zwei Wegen, die sich außerhalb des Knotens befinden. Dabei wird die Verbindung zwischen den beiden Wegen entweder nur in eine Richtung (unidirektional) oder in beide Richtungen (bidirektional) ermöglicht.

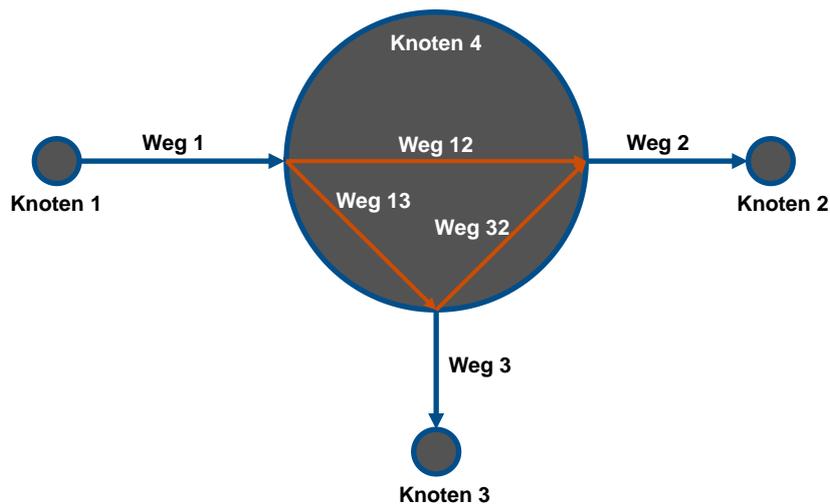


Abbildung 2-9: Modellierung von Knoten bei der BSSD

Jeder verbindende Weg wird innerhalb des `<node>`-Elements über ein untergeordnetes `<connection>`-Element repräsentiert. Dieses verfügt über das verpflichtende Attribut `connectingway`, das die `id` des verbindenden Wegs beinhaltet. Darüber hinaus existieren die verpflichtenden Attribute `incomingway` und `outgoingway`, welche die `id`-Attribute der Wege, die durch den verbindenden Weg miteinander verbunden werden, angeben. Entsprechend der Referenzrichtung des verbindenden Wegs wird einer dieser beiden Wege als eingehender Weg (`incomingway`) und der andere als ausgehender Weg (`outgoingway`) bezeichnet. Im Falle eines infinitesimalen Knotens existiert kein verbindender Weg und das Attribut `connectingway` besitzt den Wert "-1".

Für die Definition der Verbindung der Fahrstreifen des eingehenden Wegs mit den Fahrstreifen des ausgehenden Wegs besitzt ein `<connection>`-Element mindestens ein untergeordnetes `<laneLink>`-Element. Jedes `<laneLink>`-Element repräsentiert genau eine Verbindung zwischen einem

Fahrstreifen des eingehenden Wegs und einem Fahrstreifen des ausgehenden Wegs. Dazu verfügt das `<laneLink>`-Element über die drei verpflichtenden Attribute `incomingLane`, `connectingLane` und `outgoingLane`, welche die `id`-Attribute der Fahrstreifen des eingehenden, des verbindenden und des ausgehenden Wegs angeben. Der eingehende und der ausgehende Fahrstreifen ergeben sich ebenfalls durch die Referenzrichtung des verbindenden Wegs. Für den Fall eines infinitesimalen Knotens nimmt das Attribut `connectingLane` den Wert "0" an.

Zur Verdeutlichung der Verwendung der Attribute der Elemente `<connection>` und `<laneLink>` ist in Abbildung 2-10 eine beispielhafte Unterteilung der Wege des in Abbildung 2-9 gezeigten Streckennetzes in Fahrstreifen dargestellt. Die Referenzrichtung eines Wegs ist anhand der Pfeilrichtung der zu diesem Weg zugehörigen Mittellinie definiert. Die Knoten 1, 2 und 3 sind aus Gründen der Übersichtlichkeit nicht in der Abbildung enthalten. Die Ausprägungen der Attribute der zum Knoten 4 zugehörigen `<connection>`- und `<laneLink>`-Elementen sind in Tabelle 2-3 gelistet.

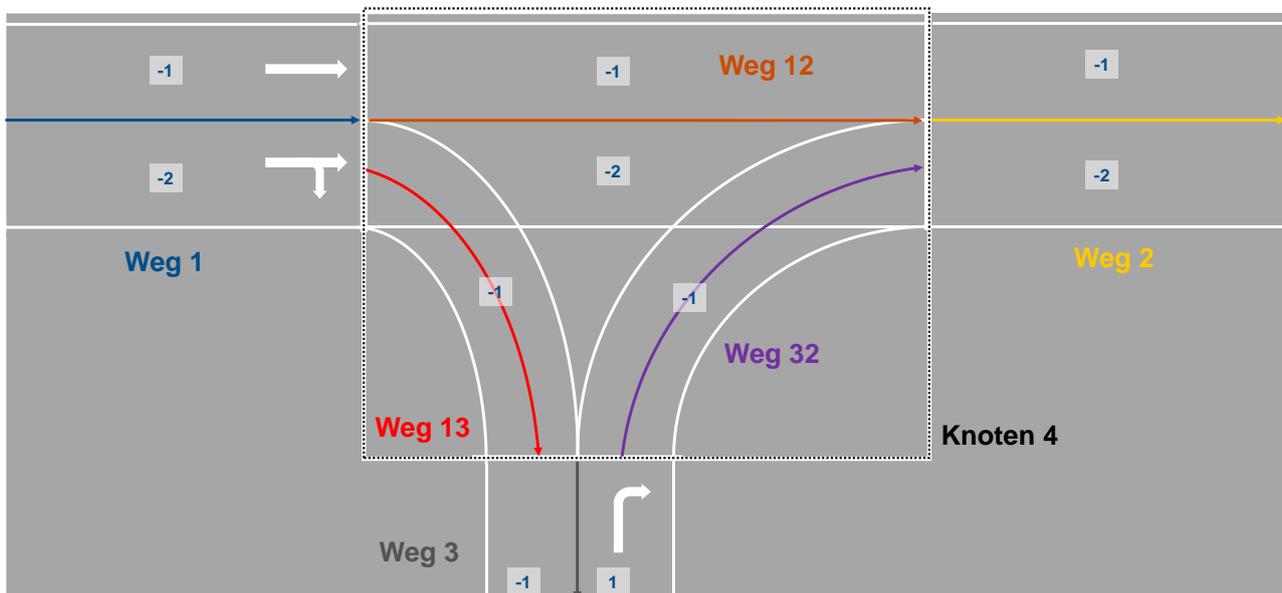


Abbildung 2-10: Unterteilung des BSSD-Streckennetzes aus Abbildung 2-9 in Fahrstreifen

Tabelle 2-3: `<connection>`- und `<laneLink>`-Elemente Knoten 4 aus Abbildung 2-10

<code>&lt;connection&gt;</code> id	<code>&lt;connection&gt;</code> incoming- Way	<code>&lt;connection&gt;</code> connecting- Way	<code>&lt;connection&gt;</code> outgoing- Way	<code>&lt;connection&gt;</code> incoming Lane	<code>&lt;connection&gt;</code> connecting- Lane	<code>&lt;connection&gt;</code> outgoing- Lane
1	1	13	3			
				-2	-1	-1
2	1	12	2			
				-1	-1	-1
				-2	-2	-2
3	2	32	3			
				1	-1	-2

---

## 3 Das OpenDRIVE-Kartenformat

---

In diesem Kapitel wird das OpenDRIVE-Kartenformat beschrieben. Dazu werden in Abschnitt 3.1 allgemeine Informationen zur Herkunft und zur Zugehörigkeit von OpenDRIVE beschrieben. In Abschnitt 3.2 werden die grundlegende Struktur sowie wichtige Definitionen und Prinzipien von OpenDRIVE erläutert. Die für das Verständnis dieser Arbeit relevanten Elemente von OpenDRIVE werden im Detail in Abschnitt 3.3 erklärt. Alle Informationen in diesem Kapitel stammen, sofern nicht anders angegeben, aus der offiziellen Spezifikation von OpenDRIVE in der Version 1.7.0<sup>39</sup>.

### 3.1 Allgemeine Informationen

OpenDRIVE ist ein offener Standard zur digitalen Beschreibung von Verkehrsnetzwerken, der primär in digitalen Simulationen in der Industrie Anwendung findet. OpenDRIVE wurde 2005 in einer Zusammenarbeit der VIRES Simulationstechnologie GmbH mit Daimler entwickelt. Seit der Veröffentlichung des Standards im Jahr 2006 beteiligen sich weitere Firmen der Automobilindustrie an der Weiterentwicklung von OpenDRIVE.<sup>40</sup> Im Jahr 2018 übernahm der aus Automobilherstellern und Zulieferern der Automobilindustrie bestehende Verein *Association for Standardisation of Automation and Measuring Systems* (ASAM e.V.)<sup>41</sup> die Verwaltung und Weiterentwicklung von OpenDRIVE.<sup>42</sup>

Eine OpenDRIVE-Karte beinhaltet neben dem eigentlichen Straßennetz weitere statische Merkmale des Verkehrsnetzwerks, wie Verkehrsschilder und statische Objekte. Somit stellt die statische Beschreibung des OpenDRIVE-Formats die Ergänzung zum ebenfalls vom ASAM e.V. verwalteten *OpenSCENARIO*-Format<sup>43</sup> dar, das eine dynamische Szenariobeschreibung ermöglicht. Weitere mit OpenDRIVE vergleichbare hochgenaue Kartenformate, die im Kontext des automatisierten Fahrens verwendet werden, sind *RoadXML*<sup>44</sup> sowie *Liblanelet*<sup>45</sup> bzw. dessen Erweiterung *Lanelet2*<sup>46</sup>.

### 3.2 Grundstruktur

Eine OpenDRIVE-Datei wird im auf der XML basierenden Dateiformat *XODR* gespeichert. Somit besteht eine OpenDRIVE-Datei analog zu einer BSSD-Datei aus mehreren hierarchisch strukturierten Elementen und einzelnen, den Elementen zugeordneten Attributen. Die hierarchische Struktur einer OpenDRIVE-Datei ist in Abbildung 3-1 gezeigt, die alle für diese Arbeit relevanten Elemente von OpenDRIVE enthält. Die in dieser Arbeit nicht verwendeten Elemente sind entweder ausgegraut (oberste Hierarchieebene) oder nicht in der Abbildung enthalten (restliche Hierarchieebenen). Analog

---

<sup>39</sup> ASAM e.V.: Specification ASAM OpenDRIVE, Version 1.7.0 (2021).

<sup>40</sup> Dupuis, M. et al.: OpenDRIVE 2010 and Beyond -Status and future of the de facto standard (2010), S. 232–233.

<sup>41</sup> ASAM e.V.: About ASAM (2021).

<sup>42</sup> MSC Software: VIRES übergibt OpenDRIVE® Standard an ASAM e.V. (2018).

<sup>43</sup> ASAM e.V.: ASAM OpenSCENARIO® (2021).

<sup>44</sup> Chaplier, J. et al.: Toward a standard: RoadXML, the road network database format (2010), S. 211–220.

<sup>45</sup> Bender, P. et al.: Lanelets: Efficient map representation for autonomous driving (2014).

<sup>46</sup> Poggenhans, F. et al.: Lanelet2: A high-definition map framework for the future of automated driving (2018).



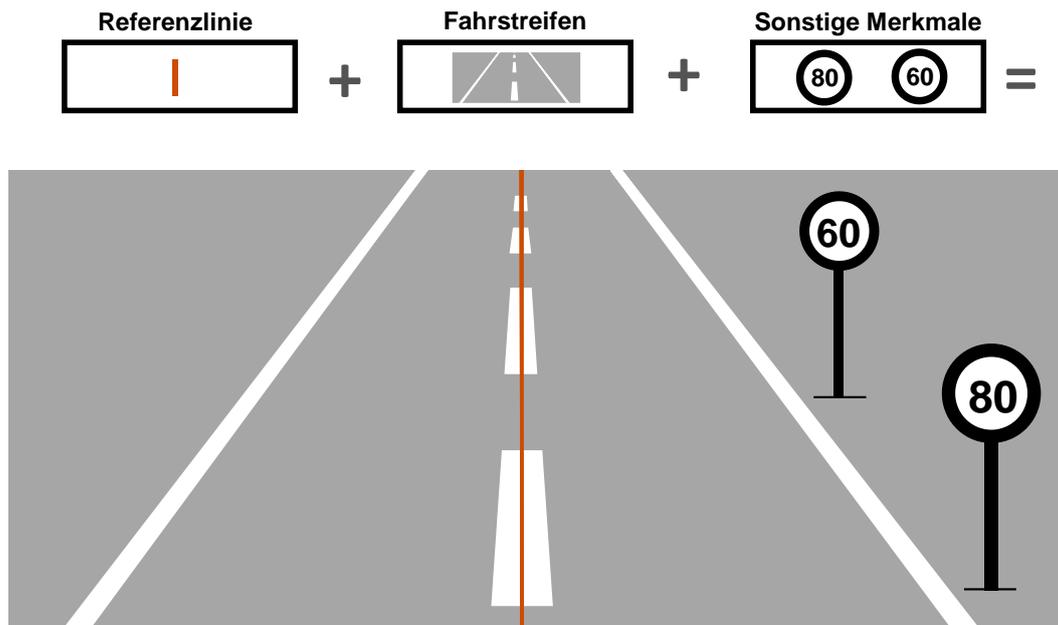


Abbildung 3-2: Zusammensetzung einer Straße in OpenDRIVE<sup>50a</sup>

Darüber hinaus sind in Abbildung 3-3 die drei in OpenDRIVE verwendeten Koordinatensysteme dargestellt. Das kartesische Inertial-Koordinatensystem  $(x, y, z)$  ist ein Rechtssystem gemäß DIN ISO 8855<sup>51</sup> und dient der Positionierung und Ausrichtung des gesamten Verkehrsnetzes einer OpenDRIVE-Karte. Hierbei ist eine Georeferenzierung möglich. Daneben existiert das ebenfalls rechtshändige Referenzlinien-Koordinatensystem  $(s, t, h)$ , das entlang der Referenzlinie einer Straße verläuft. Die  $s$ -Koordinate folgt der Tangente der Referenzlinie. Die  $t$ -Koordinate ist senkrecht zur  $s$ -Koordinate definiert. Die  $h$ -Koordinate zeigt senkrecht zur  $x,y$ -Ebene des Inertial-Koordinatensystems nach oben. Somit ist das Referenzlinien-Koordinatensystem in der  $x,y$ -Ebene des Inertial-Koordinatensystems definiert. Zuletzt existiert das Lokal-Koordinatensystem  $(u, v, z)$ , welches der Dimensionierung und Ausrichtung von Objekten dient. Das Lokal-Koordinatensystem zeigt im nicht-gedrehten Zustand entlang der Achsen des Referenzlinien-Koordinatensystems. Durch die Winkel  $\alpha$ ,  $\beta$  und  $\gamma$ , die eine Rotation um die  $u$ -,  $v$ - bzw.  $z$ -Achse angeben, werden Objekte beliebig im Raum ausgerichtet.

### 3.3 Relevante Elemente

In diesem Abschnitt werden die für das Verständnis dieser Arbeit relevanten Elemente des OpenDRIVE-Formats (siehe Abbildung 3-1) beschrieben. Dabei werden jeweils zuerst die wichtigsten dem Element zugeordneten Attribute beschrieben. Anschließend werden relevante Unterelemente beschrieben. Da die Spezifikation der BSSD in Anlehnung an das OpenDRIVE-Format definiert wurde, sind an einigen Stellen Gemeinsamkeiten in der Struktur sowie bei den verwendeten Elementen und Attributen zu erkennen.

<sup>50</sup> Eigene Darstellung nach: Dillmann, N.: ASAM OpenDRIVE Webinar, Part 1 (2020), a: S. 10; b: S.38.

<sup>51</sup> Deutsches Institut für Normung e. V.: DIN ISO 8855 (2013).

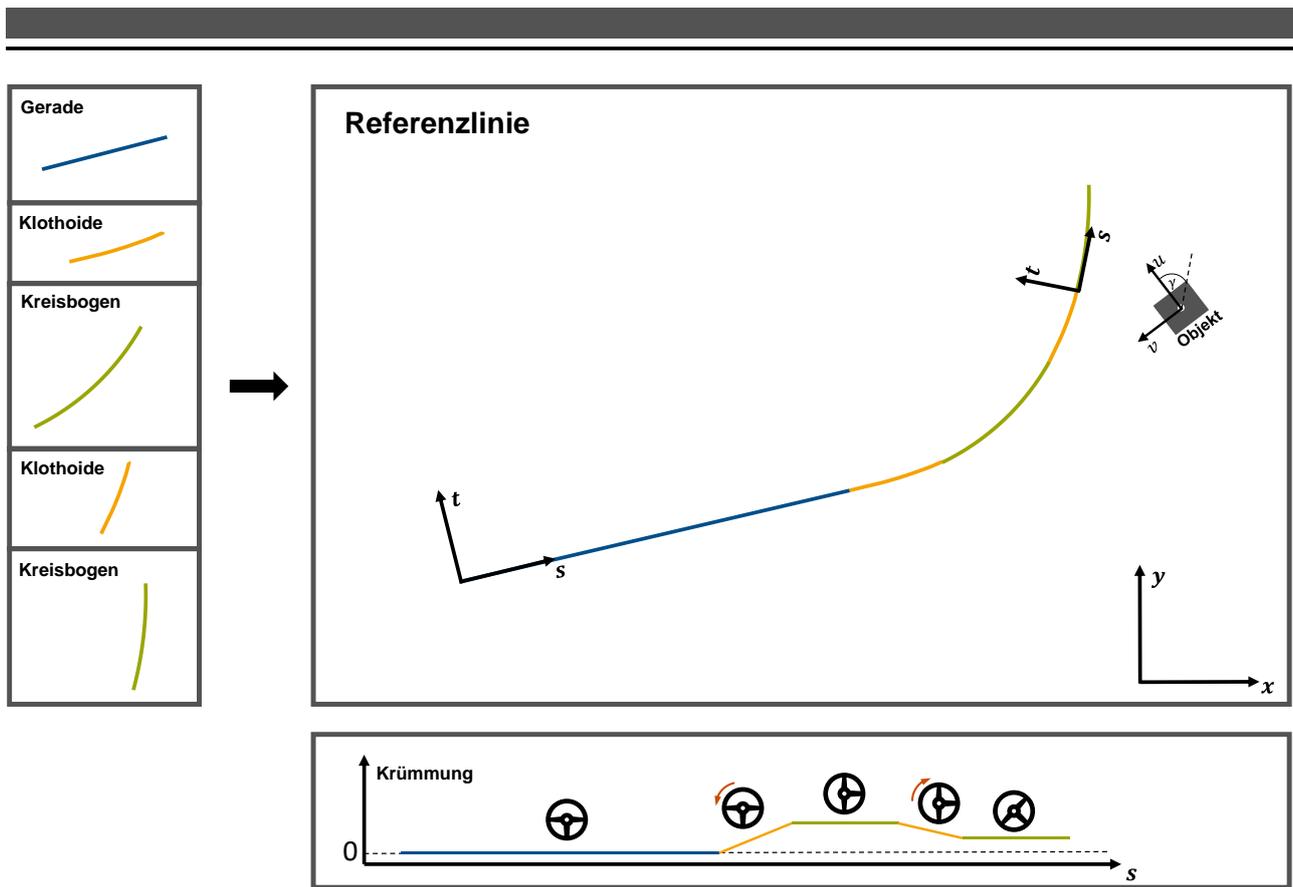


Abbildung 3-3: Referenzlinie und Koordinatensysteme in OpenDRIVE<sup>50b</sup>

### 3.3.1 Das <road>-Element

Wie zuvor beschrieben, wird eine Straße einer OpenDRIVE-Karte durch das **<road>**-Element repräsentiert. Dazu besitzt es das verpflichtende Attribut **id**, das eine eindeutige Nummer zur Identifizierung der Straße innerhalb der OpenDRIVE-Karte beinhaltet. Weiterhin existiert das verpflichtende Attribut **junction**, das die **id** der Kreuzung angibt, welche die Straße beinhaltet (siehe Abschnitt 3.3.2). Falls sich die Straße außerhalb einer Kreuzung befindet, nimmt das Attribut den Wert "-1" an. Die Länge der Straße wird im verpflichtenden Attribut **length** angegeben. Dabei wird die Länge in der Einheit Meter entlang der Referenzlinie gemessen.

#### 3.3.1.1 Das <planView>-Element

Eine Straße besitzt immer genau ein untergeordnetes **<planView>**-Element, das wiederum mindestens ein untergeordnetes **<geometry>**-Element beinhaltet. Das **<geometry>**-Element beschreibt den geometrischen Verlauf der Referenzlinie in der  $x, y$ -Ebene des Inertial-Koordinatensystems. Die Geometrieelemente Gerade, Kreisbogen und Klothoide werden durch die dem **<geometry>**-Element untergeordneten Elemente **<line>**, **<arc>** und **<spiral>** repräsentiert. Als Alternative zu den drei genannten Geometrieelementen ist die Verwendung des auf einem Polynom dritten Grades basierenden Elements **<paramPoly3>** zur Beschreibung des Straßenverlaufs möglich.

---

### 3.3.1.2 Das <link>-Element

Zum Aufbau des Straßennetzes einer OpenDRIVE-Karte ist die Verbindung einer Straße mit anderen Straßen und Kreuzungen notwendig. Dazu besitzt jede Straße, vergleichbar mit dem <way>-Element der BSSD, genau ein untergeordnetes <link>-Element. Dieses verfügt wiederum über die untergeordneten Elemente <predecessor> und <successor>, welche die der Straße vorangehende bzw. nachfolgende Straße oder Kreuzung<sup>52</sup> angeben. Hierfür besitzt ein <predecessor>- bzw. <successor>-Element das verpflichtende Attribut `elementId`, das die `id` des Vorgänger- bzw. Nachfolgerelements angibt, sowie das verpflichtende Attribut `elementType` zur Angabe des Typs des Vorgänger- bzw. Nachfolgerelements (Straße oder Kreuzung). Das optionale Attribut `contactPoint` definiert über die Ausprägungen "start" oder "end", ob der Beginn oder das Ende des Vorgänger- bzw. Nachfolgerelements an die Straße angrenzt.

### 3.3.1.3 Das <laneSection>-Element

Wie im realen Straßenverkehr besteht eine Straße in OpenDRIVE aus Fahrstreifen. Diese werden durch das dem <road>-Element untergeordneten <lanes>-Element repräsentiert. Das <lanes>-Element einer Straße besitzt mindestens ein untergeordnetes <laneSection>-Element. Eine *lane section*<sup>53</sup> repräsentiert einen in longitudinaler Richtung abgegrenzten Teil einer Straße, der eine konstante Anzahl an Fahrstreifen beinhaltet. Zur eindeutigen Identifikation einer *lane section* innerhalb einer Straße besitzt diese das verpflichtende Attribut `s`, das die `s`-Koordinate angibt bei der die *lane section* anfängt. Sobald eine weitere *lane section* mit einer höheren `s`-Koordinate definiert wird, endet die vorige *lane section* und eine neue *lane section* beginnt. Eine in zwei *lane sections* aufgeteilte Straße ist in Abbildung 3-4 dargestellt. Der Übergang von *lane section* A zu *lane section* B erfolgt sobald sich die Anzahl der Fahrstreifen von fünf auf sechs ändert.

Eine *lane section* beinhaltet die eigentlichen Fahrstreifen einer Straße, die jeweils durch ein <lane>-Element repräsentiert werden.<sup>54</sup> Jede Straße in OpenDRIVE verfügt über beliebig viele, jedoch mindestens zwei Fahrstreifen. Einer dieser beiden Fahrstreifen besitzt eine definierte Breite, um die minimale geometrische Darstellung einer Straße zu definieren. Der andere Fahrstreifen repräsentiert die Mittellinie der Straße und dient somit als Referenz für die Nummerierung der Fahrstreifen. Die Mittellinie verfügt als einziger Fahrstreifen einer *lane section* über keine Breite. Sofern für die betrachtete Straße kein Versatz der Mittellinie definiert ist (Element <laneOffset>), entspricht die Mittellinie genau der Referenzlinie der jeweiligen Straße (siehe die strichpunktierte Linie in Abbildung 3-4).

---

<sup>52</sup> Die Festlegung von Vorgänger- und Nachfolgerelement einer Straße ergibt sich durch Betrachtung der positiven `s`-Richtung

<sup>53</sup> In diesem Kapitel werden die Begriffe *lane section* und <LaneSection>-Element äquivalent verwendet

<sup>54</sup> In diesem Kapitel werden die Begriffe *Fahrstreifen* und <Lane>-Element äquivalent verwendet

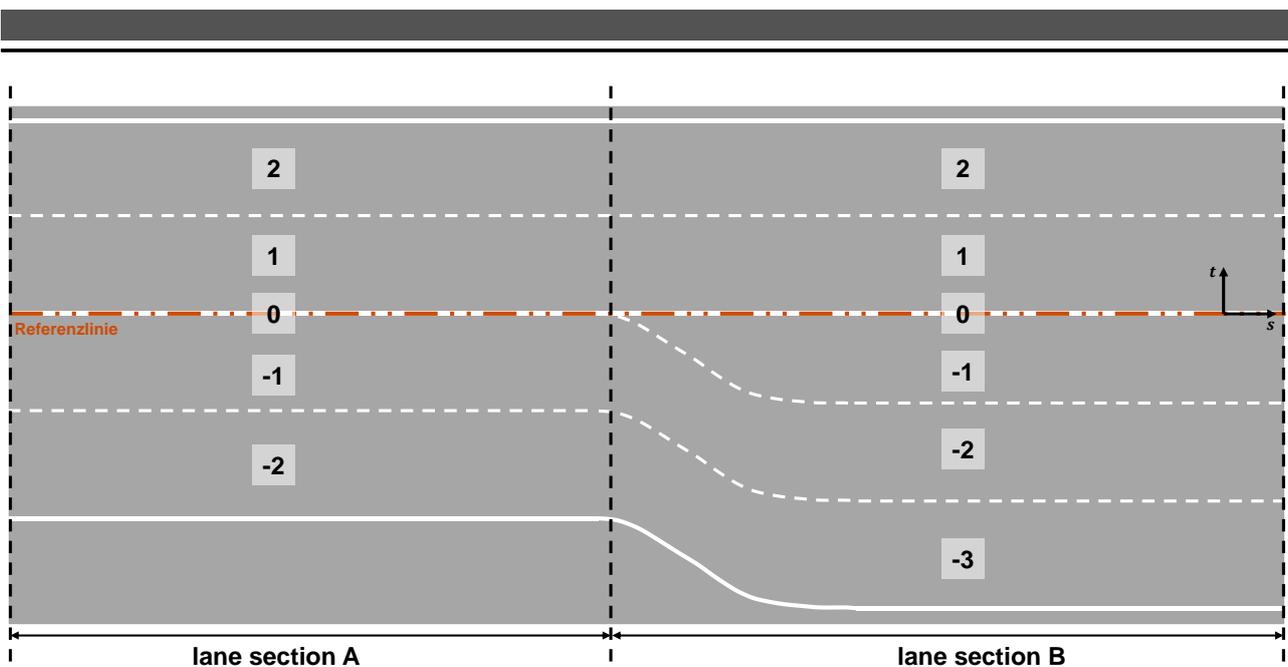


Abbildung 3-4: Aufteilung einer Straße in *lane sections* und Fahrstreifen in OpenDRIVE<sup>55</sup>

Jeder Fahrstreifen verfügt über das verpflichtende Attribut `id`, das eine eindeutige Identifikation des Fahrstreifens in der betrachteten *lane section* ermöglicht. Darüber hinaus ist anhand dieses Attributs die Bestimmung der lateralen Anordnung der Fahrstreifen innerhalb der *lane section* möglich. Ausgehend von der Mittellinie, welche immer die `id` 0 besitzt, nehmen die `id`-Werte der Fahrstreifen in negativer *t*-Richtung jeweils um den Wert Eins ab bzw. in positiver *t*-Richtung jeweils um den Wert Eins zu. Zur Verdeutlichung dieses Sachverhalts sei auf Abbildung 3-4 verwiesen. Weiterhin besitzt ein Fahrstreifen das verpflichtende Attribut `type`, das die Art des Fahrstreifens angibt (siehe Anhang A.3, Tabelle A-5). Beispielhafte Ausprägungen sind `type="driving"` („normaler“ Fahrstreifen) oder `type="sidewalk"` (Gehweg). Die in dieser Arbeit relevanten dem `<lane>`-Element untergeordneten Elemente sind in Abbildung 3-5 dargestellt und werden im Folgenden näher erläutert.

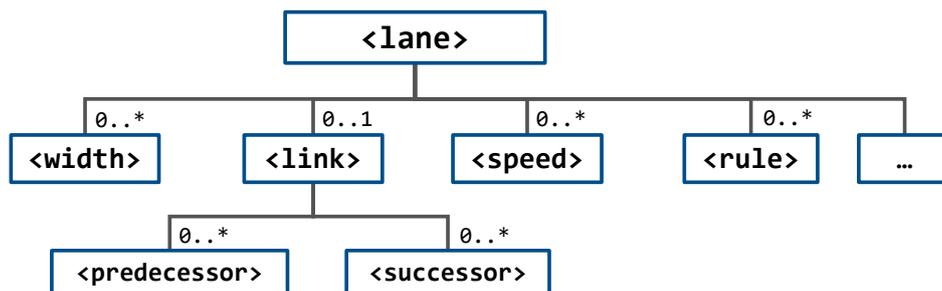


Abbildung 3-5: Hierarchische Struktur des `<lane>`-Elements in OpenDRIVE

Analog zum `<road>`-Element verfügt ein `<lane>`-Element über ein untergeordnetes `<link>`-Element, das wiederum die untergeordneten Elemente `<predecessor>` und `<successor>` besitzt. In diesem Fall dient das `<link>`-Element der Verbindung des Fahrstreifens mit einem Fahrstreifen aus der vorangehenden bzw. nachfolgenden *lane section*. Die Verbindung zum vorangehenden bzw. nachfolgenden Fahrstreifen erfolgt ebenfalls über die `id` des betreffenden Fahrstreifens. Eine

<sup>55</sup> Eigene Darstellung nach: ASAM e.V.: Specification ASAM OpenDRIVE, Version 1.7.0 (2021), Kap. 9.2.

Verbindung eines Fahrstreifens zu einem vorangehenden bzw. nachfolgenden Fahrstreifen erfolgt nur, wenn die Verbindung eindeutig ist und die beiden Fahrstreifen tatsächlich physikalisch verbunden sind. So wird beispielsweise für einen Fahrstreifen, der in einer *lane section* neu entsteht bzw. endet, kein Element `<predecessor>` bzw. `<successor>` definiert. Falls sich die vorangehende bzw. nachfolgende *lane section* in einer Kreuzung befindet, wird das Element `<predecessor>` bzw. `<successor>` nicht verwendet. In diesem Fall erfolgt die Verknüpfung der Fahrstreifen über das einer Kreuzung untergeordnete `<connection>`-Element (siehe Abschnitt 3.3.2).

Es besteht die Möglichkeit, einem Fahrstreifen eine zulässige Höchstgeschwindigkeit zuzuweisen. Dafür wird das dem `<lane>`-Element untergeordnete `<speed>`-Element verwendet. Letzteres besitzt das verpflichtende Attribut `max`, das die zulässige Höchstgeschwindigkeit in m/s angibt. Alternativ ist es möglich, die zulässige Höchstgeschwindigkeit in den Einheiten km/h oder mph anzugeben. Da es möglich ist, dass sich die zulässige Höchstgeschwindigkeit innerhalb eines Fahrstreifens mehrfach ändert, besitzt das `<speed>`-Element das Attribut `sOffset`, das die *s*-Koordinate relativ zum Beginn der dem Fahrstreifen übergeordneten *lane section* angibt, ab der die angegebene zulässige Höchstgeschwindigkeit gültig ist. Zur Verdeutlichung der Verwendung der Attribute des `<speed>`-Elements ist in Abbildung 3-6 eine aus einem Fahrstreifen bestehende Straße gezeigt. Die zulässige Höchstgeschwindigkeit beträgt zu Beginn der Straße (*s* = 0 m) 50 km/h, wird nach 200 m auf 30 km/h reduziert und bei 400 m wieder auf 50 km/h erhöht. Die aus dieser Szenerie resultierenden `<speed>`-Elemente sowie deren Attribute sind ebenfalls in Abbildung 3-6 gezeigt.

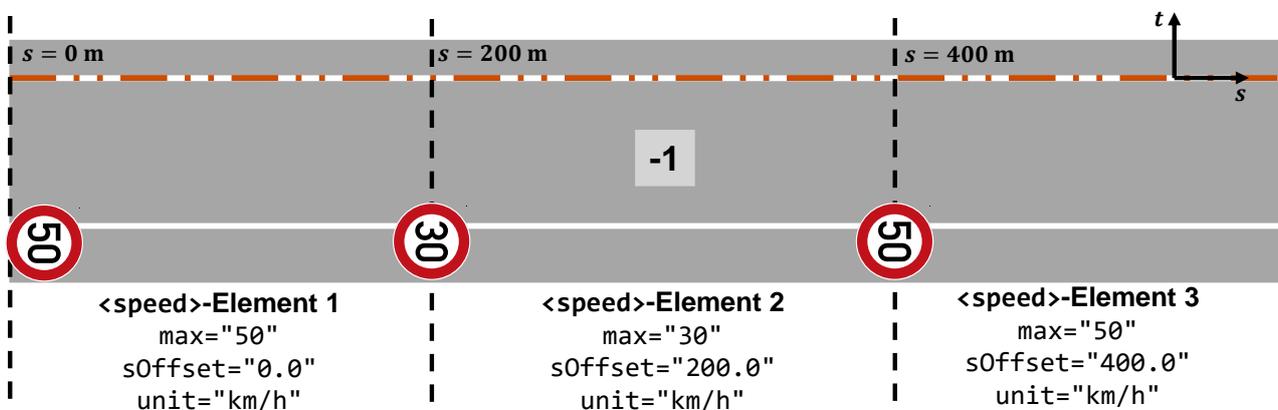


Abbildung 3-6: Verwendung des `<speed>`-Elements in OpenDRIVE

Die Breite eines Fahrstreifens wird über das dem `<lane>`-Element untergeordnete Element `<width>` definiert. Da die Möglichkeit besteht, dass sich innerhalb einer *lane section* die Breite eines Fahrstreifens mehrfach ändert, verfügt das `<width>`-Element analog zum `<speed>`-Element über das Attribut `sOffset`, das die *s*-Koordinate relativ zum Beginn der übergeordneten *lane section* angibt, ab der die im Element angegebene Breite gilt.

Zur Definition von benutzerdefinierten Verkehrsregeln für einen Fahrstreifen existiert das dem `<lane>`-Element untergeordnete `<rule>`-Element. Dieses verfügt einerseits über das verpflichtende Attribut `sOffset` zur Positionierung des Elements relativ zur *lane section* und andererseits über das verpflichtende Attribut `value`, das die benutzerdefinierte Verkehrsregel als Freitext beschreibt.

---

#### 3.3.1.4 Das <signal>-Element

OpenDRIVE bietet die Möglichkeit der Modellierung von Elementen, die im Straßenverkehr eine regulierende Funktion haben, wie beispielsweise Verkehrsschilder oder Lichtsignalanlagen. Dazu besitzt jede Straße in OpenDRIVE das optionale <signals>-Element, das wiederum über mehrere untergeordnete <signal>-Elemente verfügt. Es existieren zwei Arten von <signal>-Elementen, die über die Ausprägung des verpflichtenden Attributs `dynamic` unterschieden werden. Ein <signal>-Element mit dem Attribut `dynamic="no"` repräsentiert ein Verkehrszeichen und wird im Folgenden mit dem Begriff *statisches <signal>-Element* bezeichnet. Ein <signal>-Element mit dem Attribut `dynamic="yes"` hingegen repräsentiert ein dynamisches Element des Straßenverkehrs, das eine regulierende Funktion besitzt, wie beispielsweise eine Lichtsignalanlage oder eine Bahn-schranke. Ein <signal>-Element mit dem Attribut `dynamic="yes"` wird im Folgenden mit dem Begriff *dynamisches <signal>-Element* bezeichnet. Statische und dynamische <signal>-Elemente werden im Folgenden näher erläutert.

Ein statisches <signal>-Element repräsentiert ein Verkehrszeichen. Der Begriff Verkehrszeichen umfasst einerseits Verkehrsschilder, wie beispielsweise ein Schild zur Angabe der zulässigen Höchstgeschwindigkeit, und andererseits Fahrbahnmarkierungen, die der Regulierung des Straßenverkehrs dienen, wie beispielsweise ein Fußgängerüberweg. Für die eindeutige Identifikation eines Verkehrszeichens in der OpenDRIVE-Karte besitzt das <signal>-Element das verpflichtende Attribut `id`. Das Verkehrszeichen wird in Relation zu der das Verkehrszeichen beinhaltenden Straße platziert. Dazu besitzt das <signal>-Element die verpflichtenden Attribute `s` und `t`, welche die Position des Verkehrszeichens im Referenzlinien-Koordinatensystem der übergeordneten Straße angeben. Die im Attribut `s` angegebene `s`-Koordinate definiert zugleich den Gültigkeitsbeginn der durch das Verkehrszeichen festgelegten Regel. Falls die physische Position des Verkehrszeichens nicht mit dem Gültigkeitsbeginn der Regel übereinstimmt, wird das untergeordnete Element <positionRoad> bzw. <positionInertial> definiert. Da in jedem Land unterschiedliche Verkehrszeichen im Straßenverkehr eingesetzt werden, besitzt das <signal>-Element das optionale Attribut `country`, welches das Land, in dem das Verkehrszeichen verwendet wird, definiert. Die eindeutige Angabe eines Landes erfolgt anhand von zwei Buchstaben entsprechend der Alpha-2 Codes des Standards DIN ISO 3166-1<sup>56</sup> (z.B. „DE“ für Deutschland). Für die eindeutige Identifikation des Verkehrszeichens in dem angegebenen Land verfügt das <signal>-Element über die verpflichtenden Attribute `type` und `subtype`. Beispielsweise in Deutschland sind alle amtlich zugelassenen Verkehrszeichen der StVO im Verkehrszeichenkatalog<sup>57</sup> (VzKat) festgelegt. Im VzKat wird ein Verkehrszeichen anhand einer Hauptverkehrszeichenummer und einer Unternummer eindeutig identifiziert. Diese werden bei der Modellierung eines Verkehrszeichens aus Deutschland, was für diese Arbeit relevant ist, in OpenDRIVE in den Attributen `type` und `subtype` angegeben (siehe Abbildung 3-7 und Tabelle 3-1).

---

<sup>56</sup> Deutsches Institut für Normung e. V.: DIN ISO 3166-1: Codes für die Namen von Ländern und deren Untereinheiten (2020).

<sup>57</sup> Korsch, U.: Verkehrszeichenkatalog (VzKat) (2017).

Zur Verdeutlichung der Verwendung des `<signal>`-Elements zeigt Abbildung 3-7 beispielhaft eine Szenerie mit dem Verkehrszeichen *zulässige Höchstgeschwindigkeit* des VzKat (Hauptverkehrszeichenummer 276, Unternummer 80). Die Ausprägungen der beschriebenen Attribute des `<signal>`-Elements für das dargestellte Verkehrszeichen sind in Tabelle 3-1 gelistet.

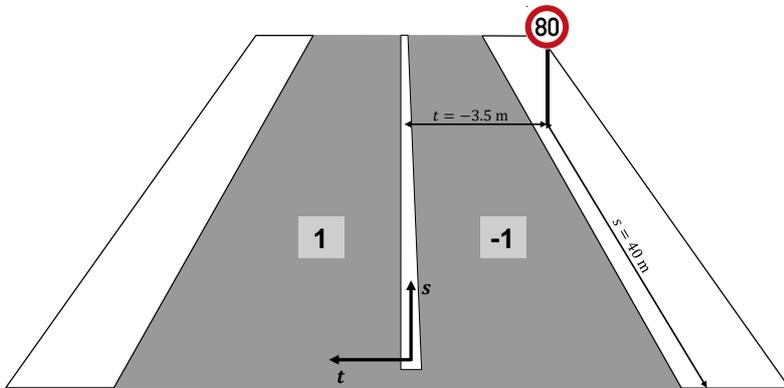


Abbildung 3-7: Szenerie mit Verkehrszeichen 274-80

Tabelle 3-1: Attribute `<signal>`-Element für Szenerie in Abbildung 3-7

Attribut	Wert
dynamic	no
s	40.0
t	-3.5
country	DE
type	274
subtype	80

Ein dynamisches `<signal>`-Element repräsentiert dynamische Elemente des Straßenverkehrs, die eine regulierende Funktion besitzen. Die Attribute eines dynamischen `<signal>`-Elements sind identisch mit den zuvor beschriebenen Attributen des statischen `<signal>`-Elements. Da dynamische Elemente des Straßenverkehrs in der Regel nicht standardisiert dokumentiert sind (siehe beispielsweise VzKat für Verkehrszeichen), werden die Attribute `type` und `subtype` bei dynamischen `<signal>`-Elementen im Allgemeinen nicht verwendet.

### 3.3.1.5 Das `<signalReference>`-Element

Da das `<signal>`-Element genau einem `<road>`-Element untergeordnet ist, besitzt die durch das `<signal>`-Element definierte Regel zunächst nur innerhalb dieser Straße Gültigkeit. Im realen Straßenverkehr ist es jedoch möglich, dass bestimmte Regeln, die in einer Straße definiert werden, über den Bereich dieser Straße hinaus gelten. Beispielsweise das Verkehrszeichen *Beginn einer Tempo 30-Zone* (Zeichen 274.1, VzKat) gilt über die dieses Verkehrszeichen beinhaltende Straße hinaus bis eine Aufhebung entsprechend des Verkehrszeichens *Ende einer Tempo 30-Zone* (Zeichen 274.2, VzKat) erfolgt. Mit einem `<signal>`-Element alleine ist diese Gegebenheit nicht modellierbar, da dieses, nur in einer Straße Gültigkeit besitzt. Um Verkehrszeichen und dynamische Elemente des Straßenverkehrs zu modellieren, deren Regel für mehrere Straßen gültig ist, existiert in OpenDRIVE das Element `<signalReference>`, das analog zum `<signal>`-Element dem Element `<signals>` untergeordnet wird. Ein `<signalReference>`-Element verweist auf ein `<signal>`-Element und repräsentiert somit den Gültigkeitsbeginn der im verwiesenen `<signal>`-Element definierten Regel in einer anderen Straße. Dazu besitzt das `<signalReference>`-Element das verpflichtende Attribut `id`, welches das `id`-Attribut des `<signal>`-Elements, auf das verwiesen wird, beinhaltet. Die `id` eines `<signal>`-Elements ermöglicht eine eindeutige Identifizierung in der gesamten OpenDRIVE-Karte (siehe Abschnitt 3.3.1.4). Weiterhin existieren die verpflichtenden Attribute `s` und `t`, die analog zum `<signal>`-Element den Gültigkeitsbeginn der durch das `<signalReference>`-Element repräsentierten Regel definieren.

### 3.3.1.6 Das <object>-Element

Weiterhin bietet OpenDRIVE die Möglichkeit der Modellierung von Objekten, die im dargestellten Verkehrsnetzwerk enthalten sind, wie beispielsweise Bäume, Pfosten, Leitplanken, Laternen oder Gebäude. Für die Modellierung der Objekte besitzt jede Straße das optionale <objects>-Element, das über mehrere untergeordnete <object>-Elemente verfügt. Ein <object>-Element repräsentiert genau ein Objekt<sup>58</sup>.

Zur eindeutigen Identifizierung eines Objekts besitzt dieses das verpflichtende Attribut `id`. Es besteht die Möglichkeit, quaderförmige und zylinderförmige Objekte zu modellieren. Dazu besitzt ein Objekt die Attribute `length`, `width` und `height` zur Festlegung der Länge, Breite und Höhe bzw. die Attribute `radius` und `height` zur Festlegung des Radius und der Höhe des Objekts. Die Angabe der Maße eines Objekts erfolgt im zu dem Objekt zugehörigen Lokal-Koordinatensystem (siehe Abschnitt 3.2), dessen Ursprung im Mittelpunkt des Objekts ist. Dabei korrespondieren die Länge, Breite und Höhe eines Objekts (Quader) mit den Koordinatenachsen  $u$ ,  $v$  und  $z$  bzw. der Radius (Zylinder) mit den Koordinatenachsen  $u$  und  $v$ . Die Ausrichtung des Objekts ergibt sich aus der Drehung des Lokal-Koordinatensystems relativ zum Referenzlinien-Koordinatensystem. Diese wird durch die optionalen Attribute `hdg`, `pitch` und `roll` definiert, welche den Gierwinkel  $\gamma$ , den Nickwinkel  $\beta$  und den Rollwinkel  $\alpha$  angeben (siehe Abbildung 3-3). Die Position eines Objektes wird anhand der verpflichtenden Attribute `s` und `t` angegeben, die den Ursprung des zu dem Objekt zugehörigen Lokal-Koordinatensystems im Referenzlinien-Koordinatensystem der jeweiligen Straße angeben. Falls sich dieser Ursprung über der Referenzlinie befindet, wird die Höhe des Ursprungs in Relation zur Höhe der Referenzlinie im verpflichtenden Attribut `zOffset` angegeben.

Abbildung 3-8 zeigt beispielhaft ein auf einer Straße befindliches zylinderförmiges und ein quaderförmiges Objekt. Dabei repräsentiert der weiße Punkt im Zentrum des jeweiligen Objekts den Ursprung des zu diesem Objekt zugehörigen Lokal-Koordinatensystems.

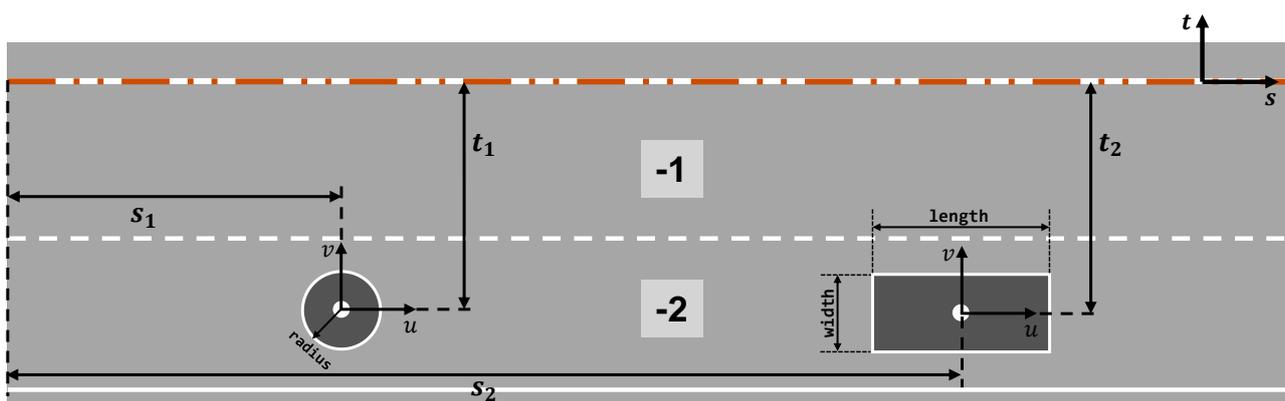


Abbildung 3-8: Zylinderförmiges und quaderförmiges Objekt in OpenDRIVE<sup>59</sup>

<sup>58</sup> In diesem Abschnitt werden die Begriffe <object>-Element und Objekt äquivalent verwendet

<sup>59</sup> Eigene Darstellung nach: ASAM e.V.: Specification ASAM OpenDRIVE, Version 1.7.0 (2021), Kap. 11.

Im Straßenverkehr existieren häufig gleichartige Objekte in unmittelbarer Umgebung zueinander, wie beispielsweise Straßenlaternen. Um die Modellierung von gleichartigen Objekten zu vereinfachen, besteht die Möglichkeit das dem `<object>`-Element untergeordnete Element `<repeat>` zu verwenden. Dabei repräsentiert ein `<repeat>`-Element mindestens ein Objekt, das gleichartig zu dem vom übergeordneten Element `<object>` repräsentierten Objekt ist. Die Anzahl der durch das `<repeat>`-Element definierten Objekte ergibt sich aus den beiden verpflichtenden Attributen `distance` und `length`. Letzteres definiert die Länge entlang der `s`-Koordinate, innerhalb der das `<repeat>`-Element definiert ist. Das Attribut `distance` gibt die Distanz zwischen den einzelnen durch das `<repeat>`-Element definierten Objekte an. Es ist möglich, den durch das `<repeat>`-Element definierten Objekten durch entsprechende Attribute eine eigene Abmessung zuzuweisen. Werden diese Attribute nicht definiert, besitzen die durch das `<repeat>`-Element definierten Objekte die gleiche Abmessung wie das übergeordnete Objekt. Die Position des ersten durch das `<repeat>`-Element repräsentierten Objekts ergibt sich analog zum `<object>`-Element durch Angabe von drei Attributen (`s`, `tStart` und `zOffset`). Die Positionen der folgenden Objekte ergeben sich aus den Attributen `length` und `distance`. Zur Verdeutlichung der Verwendung des `<repeat>`-Elements zeigt Abbildung 3-9 beispielhaft fünf Objekte, die durch ein `<repeat>`-Element definiert werden, das die dargestellten Attribute besitzt.

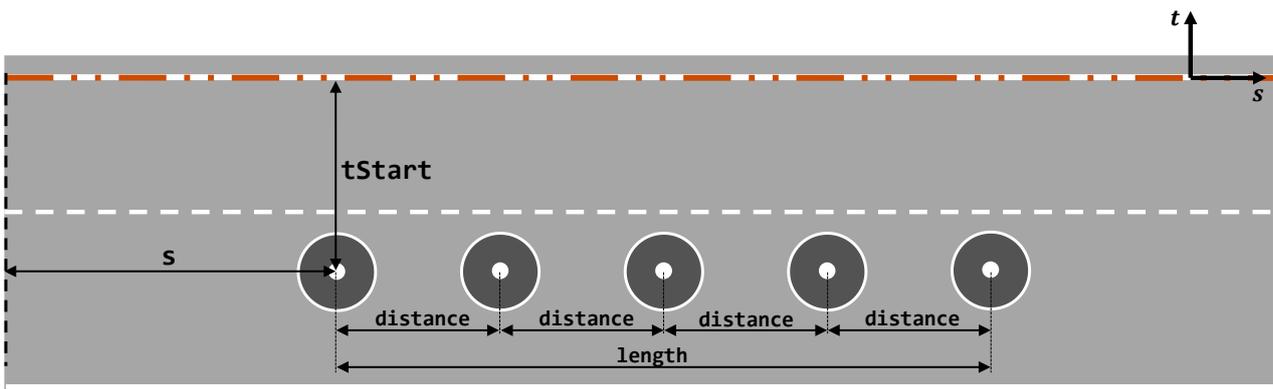


Abbildung 3-9: Objekte durch das `<repeat>`-Element in OpenDRIVE

### 3.3.2 Das `<junction>`-Element

Kreuzungen werden in OpenDRIVE durch das `<junction>`-Element repräsentiert. Sie dienen der Verbindung von drei oder mehr Straßen. Das `<junction>`-Element ist vom Aufbau und der Funktionsweise mit dem `<node>`-Element der BSSD vergleichbar. Ein `<junction>`-Element besitzt das verpflichtende Attribut `id`, welches eine eindeutige Identifikation der Kreuzung in der OpenDRIVE-Karte ermöglicht. Eine Kreuzung besteht aus mehreren Straßen, die Verbindungen zwischen den auf die Kreuzung eingehenden Straßen herstellen. Somit existieren im Kontext einer Kreuzung zwei Arten von Straßen: Die *verbindenden Straßen*, die eine Verbindung zwischen den *eingehenden Straßen* erzeugen. Eingehende Straßen sind alle Straßen, die mindestens einen Fahrstreifen beinhalten, der in Fahrtrichtung in die Kreuzung führt. Sowohl verbindende Straßen als auch eingehende Straßen werden als normale `<road>`-Elemente modelliert. Die Unterscheidung dieser beiden Arten von

Straßen ist anhand des Attributs `junction` eines `<road>`-Elements (siehe Abschnitt 3.3.1) möglich. Eingehende Straßen oder Straßen, die mit keiner Kreuzung verbunden sind, besitzen für das Attribut `junction` den Wert "-1". Verbindende Straßen beinhalten in diesem Attribut die `id` der Kreuzung zu der sie gehören.

Zur Verdeutlichung der Modellierung einer Kreuzung in OpenDRIVE ist in Abbildung 3-10 eine Kreuzung in Analogie zum BSSD-Knoten in Abbildung 2-10 dargestellt, welche die drei Straßen 1, 2 und 3 miteinander verbindet. Um die Zuordnung der Beschriftungen zu den einzelnen Straßen zu vereinfachen, sind die Referenzlinien der jeweiligen Straßen in der Farbe der zugehörigen Beschriftung eingefärbt. Die eingehenden Straßen sind in dem betrachteten Beispiel die Straßen 1 und 3. Die Straße 2 stellt keine eingehende Straße dar, weil sie keinen Fahrstreifen beinhaltet, der mit Fahrtrichtung in die Kreuzung führt. Die verbindenden Straßen sind die Straße 13 (Verbindung Straße 1 mit Straße 3), die Straße 12 (Verbindung Straße 1 mit Straße 2) und die Straße 32 (Verbindung Straße 3 mit Straße 2).

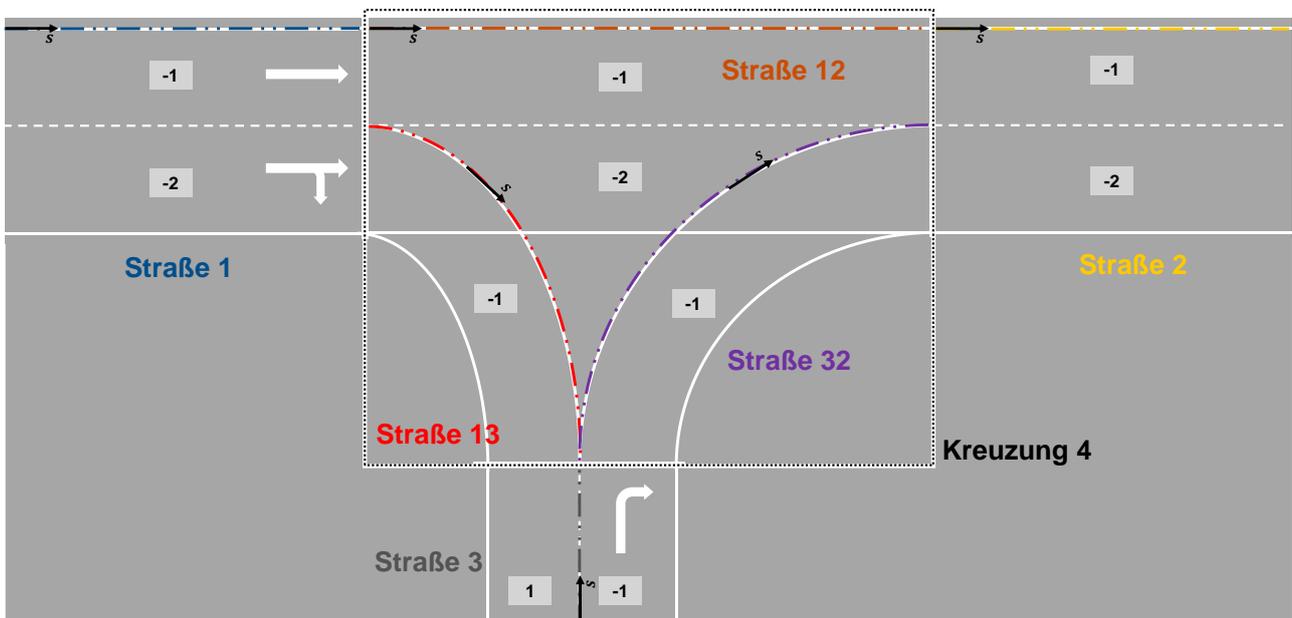


Abbildung 3-10: Beispiel einer Kreuzung in OpenDRIVE

Für jede verbindende Straße existiert, analog zu einem verbindenden Weg der BSSD, ein separates `<connection>`-Element, das dem `<junction>`-Element der zugehörigen Kreuzung untergeordnet ist. Das `<connection>`-Element besitzt das verpflichtende Attribut `id`, das eine eindeutige Identifikation des Elements innerhalb einer Kreuzung ermöglicht. Weiterhin verfügt es über die beiden optionalen Attribute `incomingRoad` und `connectingRoad`, welche die `id` der dem `<connection>`-Element zugehörigen eingehenden und verbindenden Straße angeben. Die aus der Kreuzung herausführende Straße wird nicht explizit im `<connection>`-Element referenziert. Sie ist implizit durch das `<successor>`-Element der verbindenden Straße gegeben.

Für die Definition der Verbindung der Fahrstreifen der eingehenden Straße mit den Fahrstreifen der verbindenden Straße besitzt ein `<connection>`-Element mindestens ein untergeordnetes `<laneLink>`-Element. Jedes `<laneLink>`-Element repräsentiert genau eine Verbindung zwischen einem

Fahrstreifen der eingehenden Straße und einem Fahrstreifen der verbindenden Straße. Dazu verfügt das `<laneLink>`-Element über die beiden verpflichtenden Attribute `from` und `to`, welche die `id` des Fahrstreifens in der eingehenden und der verbindenden Straße angeben. Die Verbindung der Fahrstreifen von verbindender Straße und der aus der Kreuzung herausführenden Straße wird nicht explizit im `<connection>`-Element angegeben. Sie ist implizit über das `<predecessor>`-Element des jeweiligen `<lane>`-Elements der verbindenden Straße bekannt.

Zur Verdeutlichung der praktischen Anwendung der soeben beschriebenen Modellierung einer Kreuzung sind in Tabelle 3-2 die Attribute der `<connection>`- und `<laneLink>`-Elemente der in Abbildung 3-10 dargestellten Kreuzung gelistet.

Tabelle 3-2: `<connection>`- und `<laneLink>`-Elemente Kreuzung aus Abbildung 3-10

<code>&lt;connection&gt;</code> id	<code>&lt;connection&gt;</code> incomingRoad	<code>&lt;connection&gt;</code> connectingRoad	<code>&lt;laneLink&gt;</code> from	<code>&lt;laneLink&gt;</code> to
1	1	13		
			-2	-1
2	1	12		
			-1	-1
			-2	-2
3	3	32		
			-1	-1

Es existiert der Spezialfall einer direkten Kreuzung, der vergleichbar mit einem infinitesimalen Knoten bei der BSSD ist (siehe Abschnitt 2.5.2.3). Dazu nimmt das optionale Attribut `type`, das die Art der Kreuzung angibt, die Ausprägung `"direct"` an. Bei einer direkten Kreuzung werden die in die Kreuzung eingehenden bzw. aus der Kreuzung ausgehenden Straßen direkt verbunden. Abbildung 3-11 zeigt einen Anwendungsfall einer direkten Kreuzung, der beispielsweise bei einer Autobahn-Ausfahrt Verwendung findet. Dabei teilt sich die Straße 1 in die beiden Straßen 2 und 3 auf. Da alle Fahrstreifen von Straße 1 in den jeweils nachfolgenden Fahrstreifen von Straße 2 bzw. 3 übergehen ohne dabei einen anderen Fahrstreifen zu kreuzen, wird eine direkte Kreuzung verwendet.

### 3.3.3 Das `<userData>`-Element

Das Element `<userData>` bietet die Möglichkeit, benutzerdefinierte Daten und Funktionalitäten in eine OpenDRIVE-Datei zu integrieren. Dazu besitzt es das verpflichtende Attribut `code` und das optionale Attribut `value`. Beide Attribute beinhalten einen beliebigen Freitext. Darüber hinaus besteht die Möglichkeit, dem `<userData>`-Element beliebig viele Elemente auf einer beliebigen Hierarchieebene mit einem beliebigen Namen und mit beliebigen Attributen unterzuordnen (siehe auch Abbildung 3-1). Mit dem `<userData>`-Element ist es somit möglich, hierarchiebasierte Dateistrukturen von beliebiger Tiefe in eine OpenDRIVE-Datei einzufügen.

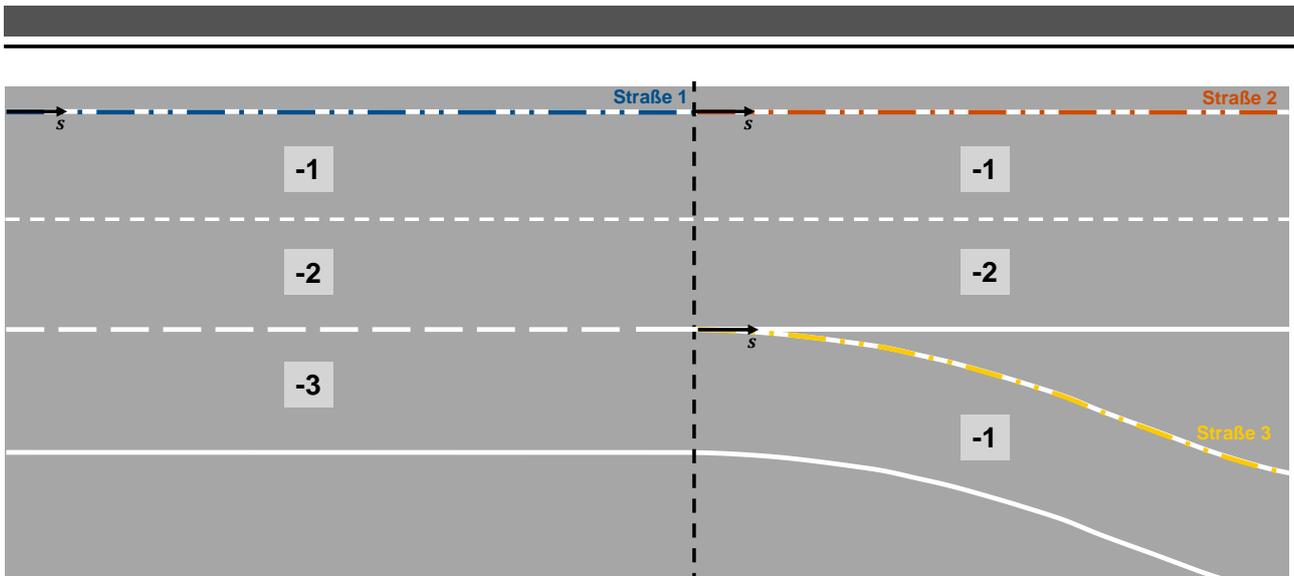


Abbildung 3-11: Beispiel einer direkten Kreuzung in OpenDRIVE<sup>60</sup>

<sup>60</sup> Eigene Darstellung nach: ASAM e.V.: Specification ASAM OpenDRIVE, Version 1.7.0 (2021), Kap. 10.4.

---

## 4 Anforderungsdefinition & -analyse

---

In diesem Kapitel werden zunächst allgemeine Anforderungen an die Integration der BSSD in ein hochgenaues Kartenformat definiert (siehe Abschnitt 4.1). Anschließend wird OpenDRIVE als ein spezielles hochgenaues Kartenformat im Hinblick auf die Erfüllung der Anforderungen analysiert (siehe Abschnitt 4.2). Diese Analyse bildet die Grundlage für die Konzeptentwicklung in Kapitel 5.

### 4.1 Ableitung der Anforderungen

Das Ziel des in dieser Arbeit zu entwickelnden Konzepts besteht darin, die ganzheitliche Integration der BSSD in eine OpenDRIVE-Karte zu ermöglichen. Um dieses Ziel zu erreichen, ergeben sich übergeordnete Anforderungen, die eine Integration der BSSD in ein hochgenaues Kartenformat erfüllen muss. Da diese Anforderungen zunächst unabhängig vom OpenDRIVE-Kartenformat sind, werden hier Anforderungen im Kontext eines beliebigen hochgenauen Kartenformats betrachtet. Da OpenDRIVE ein beispielhaftes hochgenaues Kartenformat ist, sind die formulierten Anforderungen anschließend auf OpenDRIVE anwendbar.

Damit durch die Integration der BSSD in eine hochgenaue Karte ein Mehrwert entsteht, besteht die Forderung, dass die hochgenaue Karte durch diese Integration nicht nachteilig beeinflusst wird. Dies umfasst einerseits, dass die hochgenaue Karte auch nach der Integration der BSSD für ihre ursprünglichen Zwecke verwendbar ist. Daraus resultiert die Anforderung

- (1) Die Integration der BSSD in eine Karte des hochgenauen Kartenformats darf die Nutzbarkeit der Karte nicht einschränken.**

Andererseits sollten die ursprünglich in der hochgenauen Karte enthaltenen Informationen nicht durch die Integration der BSSD verändert werden. Daraus resultiert die Anforderung

- (2) Die Integration der BSSD in eine Karte des hochgenauen Kartenformats darf zu keiner Reduktion der in der Karte vorhandenen Informationen führen im Vergleich zum Zustand der Karte vor der Integration der BSSD.**

Um das zu Beginn dieses Abschnitts beschriebene Ziel dieser Arbeit zu realisieren, leitet sich in Ergänzung zu den soeben formulierten Anforderungen die Anforderung

- (3) Die Integration der BSSD in eine Karte des hochgenauen Kartenformats muss alle Informationen der BSSD zu der Karte beinhalten**

ab. Um die vollständige Erfüllung von Anforderung (3) zu realisieren, werden untergeordnete Anforderungen definiert, die eine Karte des hochgenauen Kartenformats erfüllen muss. Diese ergeben sich aus der für die Beschreibung der BSSD minimal notwendigen Elemente und Strukturen.

Der inhaltliche Kern der BSSD wird durch die vier Verhaltensattribute gebildet, die zusammen einen atomaren Verhaltensraum definieren. In der Spezifikation der BSSD wird ein atomarer Verhaltensraum durch das `<lane>`-Element repräsentiert. Um eine Abbildung des atomaren Verhaltensraums in der hochgenauen Karte zu ermöglichen, ist erforderlich, dass die hochgenaue Karte über eine den Eigenschaften des `<lane>`-Elements entsprechende Struktur verfügt. Diese Struktur wird aufgrund dessen im Folgenden als *Fahrstreifen* bezeichnet. Die im Folgenden formulierten Anforderungen

---

leiten sich folglich aus der BSSD-Spezifikation zum <lane>-Element<sup>61</sup> ab. Zu Beginn ergibt sich die grundlegende Anforderung

**(3.1) Die Karte des Kartenformats muss basierend auf Fahrstreifen aufgebaut sein**

ab. Damit der in einem Fahrstreifen enthaltene atomare Verhaltensraum geometrisch begrenzt und eindeutig lokalisierbar ist, ergeben sich in Ergänzung zu Anforderung (3.1) die untergeordneten Anforderungen

**(3.1.1) Ein Fahrstreifen muss eindeutig identifizierbar sein,**

**(3.1.2) Ein Fahrstreifen muss lateral begrenzt sein,**

**(3.1.3) Ein Fahrstreifen muss longitudinal begrenzt sein.**

Um die laterale Anordnung der Fahrstreifen zu bestimmen, wie es in der BSSD-Spezifikation anhand des `id`-Attributs eines <lane>-Elements möglich ist, leitet sich die Anforderung

**(3.2) Einem Fahrstreifen müssen in lateraler Richtung ein linker benachbarter Fahrstreifen und ein rechter benachbarter Fahrstreifen zugewiesen werden können**

ab. Damit ein Fahrstreifen nur genau einen linken und einen rechten benachbarten Fahrstreifen besitzt, müssen lateral benachbarte Fahrstreifen eine einheitliche longitudinale Begrenzung besitzen. Daraus resultiert die untergeordnete Anforderung

**(3.2.1) Die longitudinalen Begrenzungen von in lateraler Richtung benachbarten Fahrstreifen müssen in longitudinaler Richtung identisch sein.**

Analog zur Betrachtung der lateralen Richtung erfolgt die Betrachtung der longitudinalen Richtung. Da es möglich ist, dass ein Fahrstreifen in longitudinaler Richtung mehrere vorangehende und mehrere nachfolgende Fahrstreifen besitzt, wie beispielsweise bei einem Knoten, resultiert die Anforderung

**(3.3) Einem Fahrstreifen müssen in longitudinaler Richtung mehrere vorangehende und nachfolgende Fahrstreifen zugewiesen werden können.**

Da die BSSD keine Beschränkungen bezüglich der insgesamt in lateraler und longitudinaler Richtung vorhandenen Fahrstreifen definiert, ergeben sich die beiden Anforderungen

**(3.4) Es müssen beliebig viele Fahrstreifen in lateraler Richtung aneinandergereiht werden können,**

**(3.5) Es müssen beliebig viele Fahrstreifen in longitudinaler Richtung aneinandergereiht werden können.**

Um das Attribut `crossing` der linken bzw. rechten lateralen Begrenzung eines Fahrstreifens darzustellen, resultiert die Anforderung

**(3.6) Eine laterale Begrenzung eines Fahrstreifens muss mit einer der Eigenschaften „erlaubt“, „unter Bedingungen erlaubt“, „verboten“ oder „nicht möglich“ bezüglich des Überquerens der jeweiligen Begrenzung charakterisierbar sein.**

---

<sup>61</sup> Glatzki, F.; Lippert, M.: Behavior-Semantic Scenery Description v0.4 (unveröffentlicht) (2021), S. 7–11.

---

Entsprechend der Möglichkeit, eine Bedingung für das Überqueren der linken bzw. rechten lateralen Begrenzung zu definieren (Element `<condition>`), leitet sich die untergeordnete Anforderung

**(3.6.1) Falls das Überqueren der lateralen Begrenzung mit der Eigenschaft „unter Bedingungen“ charakterisiert ist, muss eine Bedingung angegeben werden können.**

ab. In gleicher Weise ergeben sich die beiden Anforderungen

**(3.7) Die longitudinale Begrenzung zu Beginn eines Fahrstreifens muss mit einer der Eigenschaften „erlaubt“, „unter Bedingungen erlaubt“, „verboten“ oder „nicht möglich“ bezüglich des Überquerens dieser Begrenzung charakterisierbar sein,**

**(3.7.1) Falls das Überqueren der longitudinalen Begrenzung mit der Eigenschaft „unter Bedingungen“ charakterisiert ist, müssen mehrere Bedingungen angegeben werden können.**

für die longitudinale Begrenzung eines Fahrstreifens. Um die beim Element `<longitudinal>` zusätzlich vorhandenen Attribute zu berücksichtigen, die der Beschreibung verschiedener Verhaltenssätze bezüglich des Überquerens der longitudinalen Begrenzung dienen (z.B. Eintritt bei roter Lichtsignalanlage erlaubt/verboten, siehe Abschnitt 2.5.2.1), resultieren die Anforderung

**(3.8) Die longitudinale Begrenzung zu Beginn eines Fahrstreifens muss mit mehreren Eigenschaften charakterisierbar sein, die eine binäre Fallunterscheidung darstellen,**

sowie die beiden untergeordneten Anforderungen

**(3.8.1) Die binäre Fallunterscheidung „Lichtsignalanlage in Betrieb/nicht in Betrieb“ muss angegeben werden können,**

**(3.8.2) Die binäre Fallunterscheidung „Eintritt bei roter Lichtsignalanlage erlaubt/verboten“ muss angegeben werden können.**

Damit es möglich ist, das Verhaltensattribut *Reservation* in einem Fahrstreifen abzubilden, ist die Erfüllung der Anforderung

**(3.9) Einem Fahrstreifen muss eine Reservierung zugeordnet werden können**

notwendig. Da eine eindeutige Identifizierung einer Reservierung sowie die Angabe der Art der Reservierung (Attribute `id` und `type` des `<reservation>`-Elements) notwendig ist, ergeben sich die beiden untergeordneten Anforderungen

**(3.9.1) Die Reservierung muss eindeutig identifizierbar sein,**

**(3.9.2) Die Art der Reservierung muss anhand der Ausprägungen „eigen-reserviert“, „fremd-reserviert“ und „gleich-reserviert“ angegeben werden können.**

Um einer Reservierung einen oder mehrere priorisierte Verkehrsteilnehmer zuzuordnen, welche die Reservierung in Anspruch nehmen, resultiert die untergeordnete Anforderung

**(3.9.3) Der Reservierung müssen bei einer Reservierung der Art „fremd-reserviert“ oder „gleich-reserviert“ mehrere priorisierte Verkehrsteilnehmer der Klassen „Kraftfahrzeug“, „Fahrradfahrer“, „Fußgänger“ und „Schienenfahrzeug“ zugeordnet werden können.**

---

Da im Fall eines fremd- oder gleich-reservierten Fahrstreifens, die Angabe der Herkunft bzw. des Ziels des die Reservierung in Anspruch nehmenden Verkehrsteilnehmers erforderlich bzw. möglich ist, resultieren die beiden untergeordneten Anforderungen

- (3.9.3.1) Die möglichen Herkunfts-Fahrstreifen eines priorisierten Verkehrsteilnehmers müssen angegeben werden können,**
- (3.9.3.2) Die möglichen Ziel-Fahrstreifen eines priorisierten Verkehrsteilnehmers müssen angegeben werden können,**

Analog zum Element <longitudinal> besitzt das Element <reservation> mehrere Attribute, die der Beschreibung verschiedener Verhaltenssätze dienen, welche die Reservierung beeinflussen. Daraus ergeben sich die Anforderung

- (3.9.4) Die Reservierung muss mit mehreren Eigenschaften charakterisierbar sein, die eine binäre Fallunterscheidung darstellen,**

sowie die drei untergeordneten Anforderungen

- (3.9.4.1) Die binäre Fallunterscheidung „Lichtsignalanlage in Betrieb/nicht in Betrieb“ muss angegeben werden können,**
- (3.9.4.2) Die binäre Fallunterscheidung „Eintritt bei roter Lichtsignalanlage erlaubt/verboten“ muss angegeben werden können,**
- (3.9.4.3) Die binäre Fallunterscheidung „Aktive/inaktive Zusatz-Lichtsignalanlage beim Linksabbiegen“ muss angegeben werden können.**

Um das Verhaltensattribut *Speed Limit* in einem Fahrstreifen darzustellen, ergibt sich die Anforderung

- (3.10) Einem Fahrstreifen muss eine zulässige Höchstgeschwindigkeit zugewiesen werden können.**

Weiterhin ist erforderlich, dass die zulässige Höchstgeschwindigkeit über zusätzliche Bedingungen veränderbar ist, woraus die Anforderung

- (3.10.1) Die zulässige Höchstgeschwindigkeit beim Gelten zusätzlicher Bedingungen muss angegeben werden können,**

sowie die beiden untergeordneten Anforderungen

- (3.10.1.1) Die zusätzliche Bedingung eines definierten Zeitintervalls muss angegeben werden können,**
- (3.10.1.2) Die zusätzliche Bedingung einer nassen Straße muss angegeben werden können**

resultieren. Für die Repräsentation des Verhaltensattributs *Overtake* ist das Erfüllen der Anforderung

- (3.11) Einem Fahrstreifen muss eine Eigenschaft bezüglich der Erlaubnis eines Überholvorgangs zugewiesen werden können.**

notwendig.

Da entsprechend der BSSD-Spezifikation die Verhaltensattribute für beide Fahrtrichtungen innerhalb eines Fahrstreifens festgelegt werden (Elemente <behaviorAlong> und <behaviorAgainst>, siehe Abschnitt 2.5.2), ergibt sich die Anforderung

---

**(3.12) Die Anforderungen (3.6), (3.7), (3.8), (3.9), (3.10) und (3.11) müssen für beide Fahrrichtungen innerhalb eines Fahrstreifens erfüllt sein.**

Alle in diesem Abschnitt formulierten Anforderungen sind zusammengefasst im Anhang A.1, Tabelle A-1 gelistet.

## **4.2 Analyse der Anforderungsrealisierung mit OpenDRIVE**

Anhand der soeben formulierten Anforderungen wird OpenDRIVE als ein spezielles hochgenaues Kartenformat bezüglich der Realisierung dieser Anforderungen analysiert. Die Diskussion der Erfüllung der übergeordneten Anforderungen (1), (2) und (3) ist Teil des Validierungsprozesses und somit erst nach der Umsetzung eines Konzepts durchführbar (siehe Abschnitt 7.2). Aufgrund dessen beschränkt sich die im Folgenden beschriebene Analyse auf die Realisierung der Detailanforderungen (3.1) bis (3.12) durch OpenDRIVE als ein spezielles hochgenaues Kartenformat.

Das Straßennetz einer OpenDRIVE-Karte besteht in der untersten Hierarchieebene aus einzelnen Fahrstreifen (siehe Abbildung 3-1), die durch das OpenDRIVE-`<lane>`-Element<sup>62</sup> repräsentiert werden, aufgebaut. Da ein OpenDRIVE-`<lane>`-Element eine eindeutig definierte Geometrie besitzt sowie anhand des zugehörigen Attributs `id` eindeutig identifizierbar ist, erfüllt OpenDRIVE die Anforderung (3.1) („Die Karte des Kartenformats muss basierend auf Fahrstreifen aufgebaut sein“) sowie die untergeordneten Anforderungen (3.1.1), (3.1.2) und (3.1.3), die fordern, dass ein Fahrstreifen eindeutig identifizierbar sowie lateral und longitudinal begrenzt ist. Wie in Abschnitt 3.3.1.3 beschrieben, ist anhand des Attributs `id` eines OpenDRIVE-Fahrstreifens die laterale Anordnung der OpenDRIVE-Fahrstreifen einer *lane section* bestimmbar. Weiterhin besitzen lateral benachbarte OpenDRIVE-Fahrstreifen durch die Einteilung einer Straße in *lane sections* identische longitudinale Begrenzungen. Aufgrund dessen erfüllt OpenDRIVE die Anforderung (3.2) („Einem Fahrstreifen müssen in lateraler Richtung ein linker benachbarter Fahrstreifen und ein rechter benachbarter Fahrstreifen zugewiesen werden können“) sowie die daraus resultierende Anforderung (3.2.1), die eine einheitliche longitudinale Begrenzung von lateral benachbarten Fahrstreifen fordert. Ein OpenDRIVE-Fahrstreifen wird in longitudinaler Richtung mit beliebig vielen vorangehenden bzw. nachfolgenden OpenDRIVE-Fahrstreifen verknüpft. Dies erfolgt entweder über die dem OpenDRIVE-`<lane>`-Element untergeordneten `<predecessor>`- bzw. `<successor>`-Elemente oder über die `<laneLink>`-Elemente einer Kreuzung (siehe Abschnitt 3.3.1.6). Folglich erfüllt OpenDRIVE die Anforderung (3.3) („Einem Fahrstreifen müssen in longitudinaler Richtung mehrere vorangehende und nachfolgende Fahrstreifen zugewiesen werden können“). Da OpenDRIVE keine Beschränkungen bezüglich der Anzahl der in lateraler und longitudinaler Richtung aneinandergereihten OpenDRIVE-Fahrstreifen vorgibt, sind die Anforderungen (3.4) und (3.5) ebenfalls erfüllt.

Die Anforderungen (3.6), (3.7), (3.8), (3.9), (3.10) und (3.11) sowie die zu diesen Anforderungen untergeordneten Anforderungen fordern zusammengefasst, dass innerhalb eines Fahrstreifens des

---

<sup>62</sup> Aufgrund der identischen Bezeichnung eines Fahrstreifens als `<Lane>-Element` bei OpenDRIVE und bei der BSSD werden im Folgenden die Begriffe *OpenDRIVE-`<Lane>-Element`* bzw. *OpenDRIVE-Fahrstreifen* und *BSSD-`<Lane>-Element`* bzw. *BSSD-Fahrstreifen* zur eindeutigen Unterscheidung verwendet

---

hochgenauen Kartenformats ein atomarer Verhaltensraum der BSSD ableger ist. Da die BSSD nicht im Standard OpenDRIVE enthalten ist, verfügt OpenDRIVE über keine vordefinierten Strukturen, um die Informationen zu einem atomaren Verhaltensraum der BSSD abzuspeichern. Somit erfüllen die in der Spezifikation von OpenDRIVE vordefinierten Elemente im Allgemeinen die Anforderungen (3.6) bis (3.11) nicht. Es sei angemerkt, dass die einem OpenDRIVE-`<lane>`-Element untergeordneten Elemente `<speed>` und `<rule>` (siehe Abschnitt 3.3.1.3) eine Ausnahme zu dieser Aussage darstellen. Diese beiden Elemente erfüllen die Anforderungen (3.10) und (3.11), welche fordern, dass einem Fahrstreifen eine zulässige Höchstgeschwindigkeit und eine Eigenschaft bezüglich der Erlaubnis eines Überholvorgangs zuweisbar sein muss. Jedoch erlauben die Elemente `<speed>` und `<rule>` keine Realisierung der Anforderung (3.12), die eine Erfüllung der Anforderungen (3.10) und (3.11) für beide Fahrtrichtungen innerhalb eines Fahrstreifens fordert. Aufgrund dessen ist die Verwendung der Elemente `<speed>` und `<rule>` nicht zielführend.

Wie in Abschnitt 3.3.3 beschrieben, besteht die Möglichkeit, OpenDRIVE mit dem `<userData>`-Element um benutzerdefinierte Elemente und Attribute mit beliebiger Strukturtiefe zu erweitern. Aufgrund dessen ist eine Realisierung der dem atomaren Verhaltensraum der BSSD zugehörigen Anforderungen (3.6) bis (3.12) anhand einer benutzerdefinierten Erweiterung des OpenDRIVE-Standards möglich. Die Entwicklung einer benutzerdefinierten Erweiterung mit dem `<userData>`-Element zur Abbildung des Verhaltensraums ist der Haupt-Bestandteil der in Kapitel 5 folgenden Konzeptentwicklung.

---

## 5 Konzeptentwicklung

---

In diesem Kapitel wird ein Konzept entwickelt, das eine ganzheitliche Integration der BSSD in eine vorhandene OpenDRIVE-Karte entsprechend der im vorigen Kapitel formulierten Detailanforderungen ermöglicht. Dazu werden in Abschnitt 5.1 ausgehend von den hierarchischen Strukturen von OpenDRIVE und der BSSD mehrere Konzepte abgeleitet, deren Umsetzung eine Integration der BSSD in OpenDRIVE prinzipiell ermöglichen. Die abgeleiteten Konzepte werden in Abschnitt 5.2 anhand von gewichteten Bewertungskriterien verglichen, sodass das Konzept mit dem höchsten Gesamtnutzen für die Umsetzung ausgewählt wird. In Abschnitt 5.3 wird das ausgewählte Konzept in einzelne zur Umsetzung notwendige Schritte zerlegt und bezüglich einer Umsetzung bewertet.

### 5.1 Ableitung von Konzepten

Die vollständige Integration der BSSD in eine OpenDRIVE-Karte ist dann erfolgt, wenn entsprechend Anforderung (3) (Abschnitt 4.1) alle BSSD-Informationen zu der von der OpenDRIVE-Datei repräsentierten Karte in dieser Karte gespeichert sind. Der Ausgangspunkt für die Ableitung der Konzepte ist das Ergebnis der Analyse des OpenDRIVE-Kartenformats bezüglich der Realisierung der Anforderungen an eine Integration der BSSD in ein hochgenaues Kartenformat (siehe Abschnitt 4.2). Es hat sich gezeigt, dass OpenDRIVE grundsätzliche Anforderungen für eine Integration der BSSD erfüllt, jedoch keine Möglichkeit bietet, den Verhaltensraum in der vordefinierten OpenDRIVE-Struktur abzulegen. Da OpenDRIVE anhand des Elements `<userData>` um benutzerdefinierte Strukturen erweiterbar ist, ist es dennoch möglich, den Verhaltensraum in die OpenDRIVE-Struktur zu integrieren. Somit ist das Ziel der im Folgenden abgeleiteten Konzepte, eine benutzerdefinierte Struktur zu definieren, die eine vollständige Speicherung der BSSD-Informationen in der OpenDRIVE-Karte ermöglicht.

Als Grundlage für die Ableitung der Konzepte zeigt Abbildung 5-1 die hierarchischen Strukturen von OpenDRIVE (links) und BSSD (rechts) im Vergleich. Die Strukturen sind in vereinfachter Form dargestellt, da sie nur die für die Konzeptableitung wesentlichen Elemente der jeweiligen Formate enthalten (Vgl. Abbildung 2-7 und Abbildung 3-1). Das im linken Zweig der BSSD-Struktur gezeigte Element *Verhaltensraum* repräsentiert die in einem BSSD-`<lane>`-Element befindlichen Elemente `<behaviorAlong>` und `<behaviorAgainst>` sowie die zu diesen Elementen untergeordneten Elemente.

Im Folgenden werden drei Konzepte beschrieben, die sich aus der Integration von Teilen der BSSD-Struktur bzw. der kompletten BSSD-Struktur in unterschiedliche Hierarchiestufen der OpenDRIVE-Struktur ableiten.

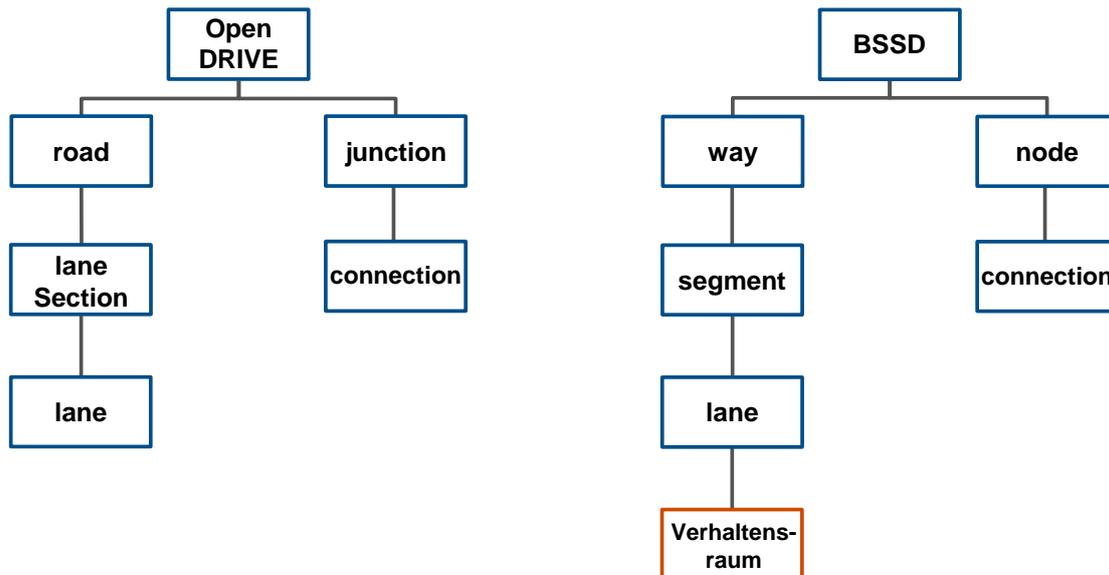


Abbildung 5-1: Vergleich der Strukturen von OpenDRIVE und BSSD

### 5.1.1 Konzept 1: <lane>-Ebene

Das Grundprinzip des ersten Konzepts ist in Abbildung 5-2 dargestellt. Die Idee dieses Konzepts besteht darin, den Verhaltensraum direkt in einen OpenDRIVE-Fahrstreifen der OpenDRIVE-Karte zu integrieren. Aufgrund dessen besitzt dieses Konzept den Namen „<lane>-Ebene“.

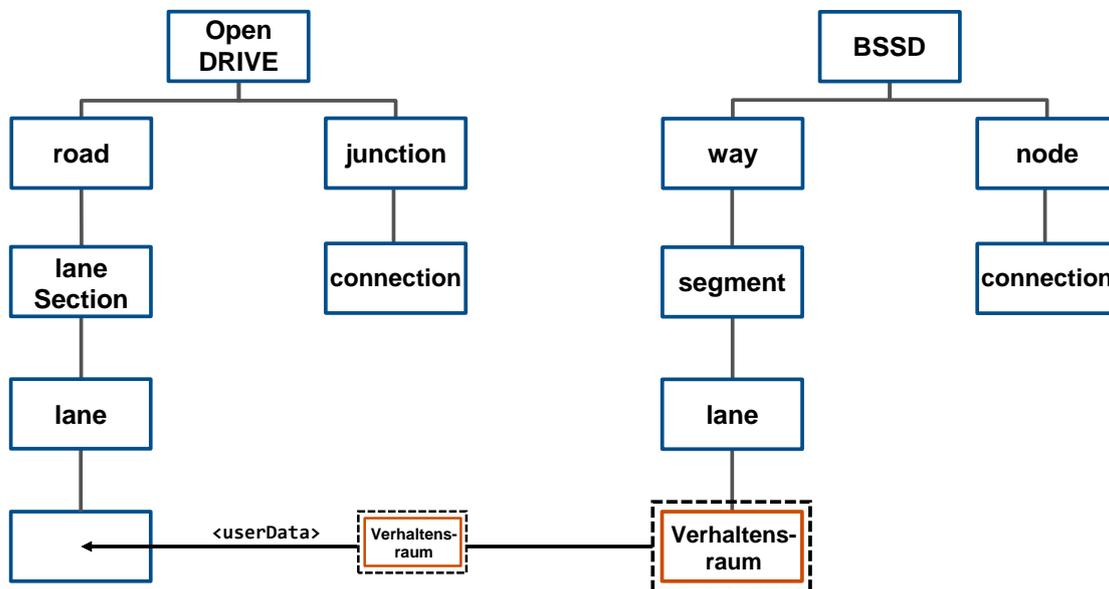


Abbildung 5-2: Grundprinzip Konzept 1, <lane>-Ebene

Da der Verhaltensraum der BSSD bei diesem Konzept direkt in der untersten strukturellen Ebene des OpenDRIVE-Formats eingefügt wird, wäre die Bestimmung des Verhaltensraums für eine bestimmte Position in der OpenDRIVE-Karte einfach und unkompliziert möglich. Für die Realisierung dieses Konzepts wäre es erforderlich, die OpenDRIVE-Fahrstreifen in longitudinaler Richtung entsprechend der longitudinalen Begrenzungen der BSSD-Verhaltensräume anzupassen. Folglich wäre es

notwendig, an jeder Stelle der OpenDRIVE-Karte, bei der ein BSSD-Segment entsprechend der Ausprägungen der BSSD-Verhaltensattribute definiert wird, ebenfalls eine neue *lane section* zu definieren. Daraus folgt, dass ein Aufteilen bzw. ein Zusammenfügen der bestehenden *lane sections* einer OpenDRIVE-Karte notwendig wäre.

Zur Verdeutlichung dieses Sachverhalts wird auf die in Abbildung 5-3 dargestellte Straße verwiesen. Die resultierende longitudinale Unterteilung der Straße in *lane sections* und Segmente ist anhand der gestrichelten Linien gezeigt. Dabei repräsentieren die schwarzen Linien die Begrenzungen der *lane sections*, während die blauen Linien die Begrenzungen der BSSD-Segmente definieren. In der abgebildeten Straße bildet sich bei einer *s*-Koordinate von 20 m ein zusätzlicher OpenDRIVE-Fahrstreifen. Dementsprechend beginnt bei dieser Position sowohl eine neue *lane section* als auch ein neues BSSD-Segment. Bei einer *s*-Koordinate von 50 m ändert sich die zulässige Höchstgeschwindigkeit durch ein Verkehrszeichen auf 60 km/h. Somit ändert sich das BSSD-Verhaltenstribut *Speed Limit* in longitudinaler Richtung und die Definition eines neuen BSSD-Segments ist erforderlich. Die geänderte zulässige Höchstgeschwindigkeit hat keinen Einfluss auf die Einteilung der Straße in *lane sections*. Wie das Beispiel zeigt, existieren durch die *lane sections* in OpenDRIVE einerseits und durch die BSSD-Segmente andererseits zwei separate longitudinale Unterteilungen einer Straße der OpenDRIVE-Karte. Diese beiden Unterteilungen sind nur zu Teilen identisch. Folglich ist eine Anpassung der *lane sections* an die BSSD-Segmente notwendig. Im Beispiel aus Abbildung 5-3 wäre es somit erforderlich, bei einer *s*-Koordinate von 50 m eine dritte *lane section* zu definieren. Dazu wäre ein Aufteilen der vorhandenen *lane section* B notwendig. Darüber hinaus ist auch das umgekehrte Szenario denkbar, bei dem im Definitionsbereich mehrerer *lane sections* nur ein BSSD-Segment definiert ist (siehe Abbildung 5-5). Aufgrund dessen wäre das Zusammenfügen mehrerer bestehender *lane sections* zu einer *lane section* erforderlich.

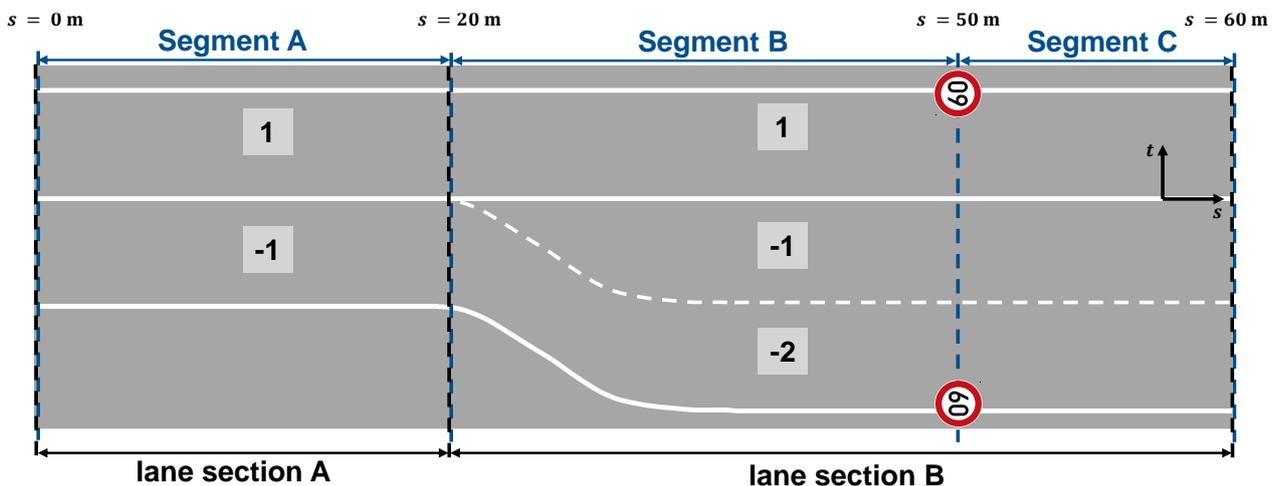


Abbildung 5-3: Straße mit unterschiedlicher Einteilung von *lane sections* und BSSD-Segmenten, Beispiel 1

### 5.1.2 Konzept 2: <road>-Ebene

Das Grundprinzip des zweiten Konzepts ist in Abbildung 5-4 dargestellt. Die Idee dieses Konzepts besteht darin, die zu einer Straße zugehörigen BSSD-Segmente sowie die zu diesen BSSD-Segmenten untergeordneten BSSD-Fahrstreifen mit den einzelnen atomaren Verhaltensräumen, in eine

Straße von OpenDRIVE zu integrieren. Aufgrund dessen besitzt dieses Konzept den Namen „<road>-Ebene“. Im Unterschied zu Konzept 1 (<lane>-Ebene) wird die longitudinale Begrenzung der jeweiligen atomaren Verhaltensräume bei diesem Konzept nicht über die vorhandenen *lane sections*, sondern über eine separate Struktur, die BSSD-Segmente, definiert. Folglich ist im Gegensatz zu Konzept 1 keine Anpassung der bereits vorhandenen *lane sections* in OpenDRIVE notwendig.

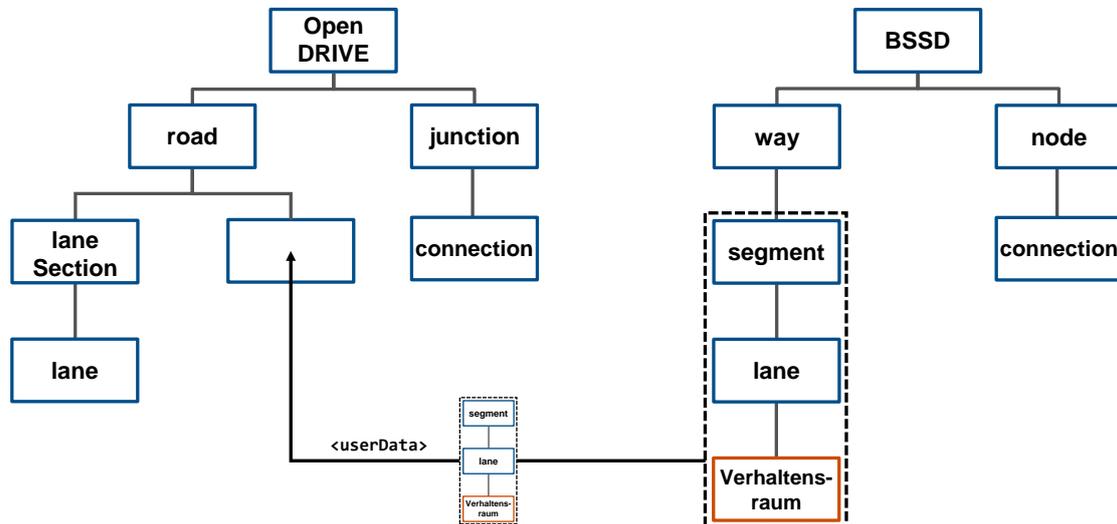


Abbildung 5-4: Grundprinzip Konzept 2, <road>-Ebene

Durch die *lane sections* einerseits und die BSSD-Segmente andererseits resultieren bei diesem Konzept zwei separate longitudinale Aufteilungen einer Straße. Daraus ergibt sich, dass zwei separate Arten von Fahrstreifen existieren. Einerseits die OpenDRIVE-Fahrstreifen (OpenDRIVE-<lane>-Elemente), welche die geometrischen Informationen des Straßennetzes beinhalten. Andererseits die BSSD-Fahrstreifen (BSSD-<lane>-Elemente), welche die zu dem Straßennetz zugehörigen atomaren Verhaltensräume beinhalten. Die longitudinalen Begrenzungen der OpenDRIVE- und der BSSD-Fahrstreifen sind aufgrund der unterschiedlichen Begrenzungen der *lane sections* und der BSSD-Segmente im Allgemeinen nicht einheitlich. Folglich ist es nicht möglich, einem OpenDRIVE-Fahrstreifen genau einen BSSD-Fahrstreifen zuzuordnen und umgekehrt. Für die Bestimmung des an einer bestimmten Stelle der OpenDRIVE-Karte geltenden Verhaltensraums ist somit eine eindeutige Zuordnung zwischen den OpenDRIVE- und BSSD-Fahrstreifen für eine definierte Position der Straße notwendig.

Zur Verdeutlichung dieses Sachverhalts wird auf die in Abbildung 5-5 dargestellte Straße verwiesen. Die Straße ist in einen Bereich für Kraftfahrzeuge (oberhalb des Grünstreifens) und einen Bereich für Fußgänger und Fahrradfahrer (unterhalb des Grünstreifens) aufgeteilt. Analog zu Abbildung 5-3 ist die resultierende longitudinale Unterteilung der Straße in *lane sections* und BSSD-Segmente dargestellt. Ein OpenDRIVE- bzw. BSSD-Fahrstreifen ist anhand der Schriftfarbe schwarz bzw. blau der übergeordneten *lane section* bzw. dem übergeordneten BSSD-Segment zuordenbar. Bei einer *s*-Koordinate von 30 m teilt sich der abgebildete Geh- und Radweg, der durch den OpenDRIVE-Fahrstreifen mit der *id* -3 repräsentiert wird, in zwei separate OpenDRIVE-Fahrstreifen (*id* -3 und -4) auf. Dementsprechend wird in OpenDRIVE eine neue *lane section* definiert. Da die BSSD nur für die OpenDRIVE-Fahrstreifen, die ein Teil der Fahrbahn sind, definiert wird, wird der gesamte Bereich

für Fußgänger und Radfahrer nicht mit der BSSD modelliert. Folglich wird bei einer  $s$ -Koordinate von 30 m kein neues BSSD-Segment definiert. Bei einer  $s$ -Koordinate von 50 m ändert sich die zulässige Höchstgeschwindigkeit durch ein Verkehrszeichen auf 60 km/h. Analog zu Abbildung 5-3 resultiert daraus die Definition eines neuen BSSD-Segments, jedoch nicht die Definition einer neuen *lane section*.

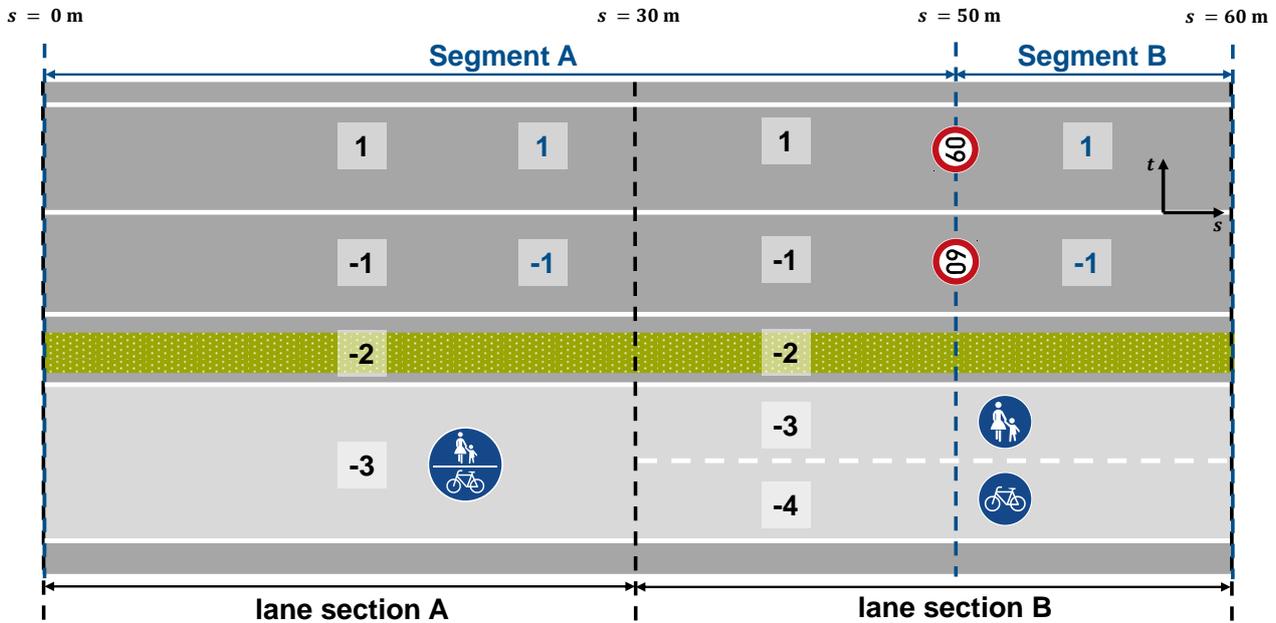


Abbildung 5-5: Straße mit unterschiedlicher Einteilung von *lane sections* und BSSD-Segmenten, Beispiel 2

Es zeigt sich, dass im Allgemeinen keine eindeutige Zuordnung eines BSSD-Fahrstreifens zu genau einem OpenDRIVE-Fahrstreifen möglich ist. Beispielsweise der BSSD-Fahrstreifen mit der *id* 1 aus dem Segment A entspricht im Bereich der  $s$ -Koordinate von 0 m bis 30 m dem OpenDRIVE-Fahrstreifen mit der *id* 1 aus *lane section A* und im Bereich der  $s$ -Koordinate von 30 m bis 50 m dem OpenDRIVE-Fahrstreifen mit der *id* 1 aus *lane section B*. Die eindeutige Zuordnung zwischen einem BSSD-Fahrstreifen und einem OpenDRIVE-Fahrstreifen ist folglich nur für eine definierte  $s$ -Koordinate der Straße möglich. Die Definition einer Struktur zur eindeutigen Zuordnung zwischen OpenDRIVE- und BSSD-Fahrstreifen ist somit eine notwendige Voraussetzung für die Realisierung dieses Konzepts.

### 5.1.3 Konzept 3: <OpenDRIVE>-Ebene

Das Grundprinzip des dritten Konzepts ist in Abbildung 5-6 dargestellt. Die Idee dieses Konzepts besteht darin, die komplette BSSD-Struktur getrennt zu den in der OpenDRIVE-Karte bereits vorhandenen Straßen und Kreuzungen zu integrieren. Aufgrund dessen wird dieses Konzept als „<OpenDRIVE>-Ebene“ bezeichnet. Bei diesem Konzept werden, im Gegensatz zu Konzept 2 (<road>-Ebene), die die Fahrstreifen des jeweiligen Formats beinhaltenden Elemente (*lane section* bzw. BSSD-Segment) in separaten Strukturen (Elemente <road> und <way>) definiert. Aufgrund dessen ist für die gewünschte Bestimmung der BSSD-Verhaltensattribute an einer definierten Position der OpenDRIVE-Karte in einem ersten Schritt eine eindeutige Zuordnung zwischen einer

OpenDRIVE-Straße und einem BSSD-Weg notwendig. In einem zweiten Schritt ist, analog zu Konzept 2 (<road>-Ebene), erforderlich, für eine definierte Position der OpenDRIVE-Karte eine eindeutige Zuordnung zwischen den OpenDRIVE- und BSSD-Fahrstreifen herzustellen. Die Definition einer Struktur, welche die Durchführung dieser beiden Schritte ermöglicht, ist daher eine notwendige Voraussetzung für die Realisierung dieses Konzepts.

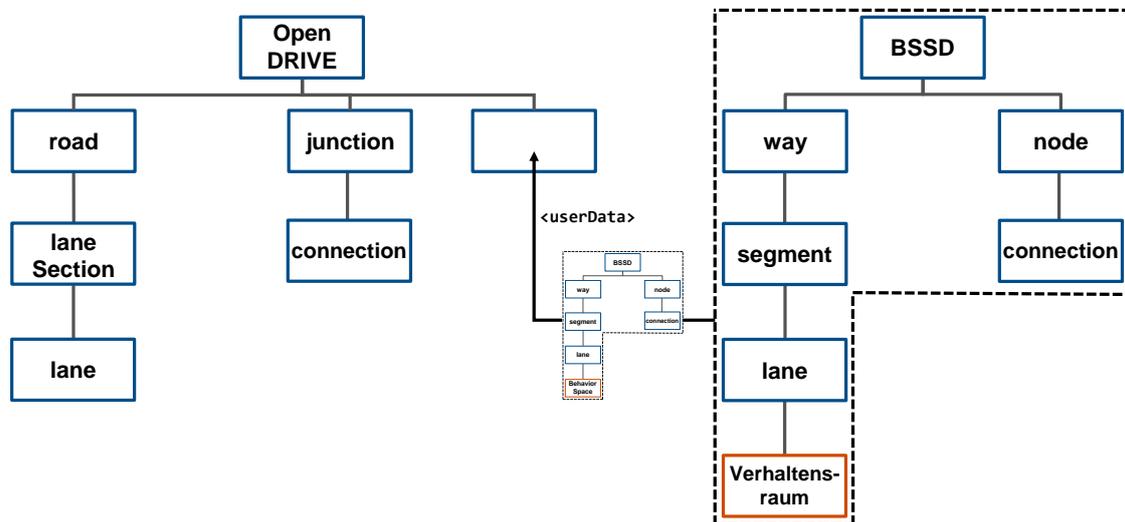


Abbildung 5-6: Grundprinzip Konzept 3, <OpenDRIVE>-Ebene

## 5.2 Auswahl eines Konzepts

In diesem Abschnitt wird eines der in Abschnitt 5.1 abgeleiteten Konzepte für eine Umsetzung ausgewählt. Für einen Vergleich der Konzepte werden in Abschnitt 5.2.1 zunächst Bewertungskriterien definiert und bezüglich ihrer Relevanz für die in dieser Arbeit umzusetzende BSSD-Integration bewertet. Basierend auf den Einschätzungen der Relevanz der Bewertungskriterien wird in Abschnitt 5.2.2 eine Gewichtung der Bewertungskriterien anhand der Methode des Paarvergleichs vorgenommen. In Abschnitt 5.2.3 werden die drei abgeleiteten Konzepte anhand eines Bewertungsschemas auf die definierten Bewertungskriterien angewendet. Anhand der Gewichtung der Bewertungskriterien sowie der Anwendung dieser auf die Konzepte erfolgt in Abschnitt 5.2.4 ein quantitativer Vergleich der drei Konzepte mittels einer Nutzwertanalyse. Nach abschließender Überprüfung der Robustheit des Ergebnisses der Nutzwertanalyse sowie der Betrachtung möglicher inhaltlicher Ursachen für das Ergebnis wird ein Konzept für eine Umsetzung ausgewählt.

### 5.2.1 Definition von Bewertungskriterien

Im Folgenden werden Bewertungskriterien definiert, die dem Vergleich der abgeleiteten Konzepte dienen. Die Bewertungskriterien ergeben sich durch Betrachtung der charakteristischen Merkmale und Unterschiede der abgeleiteten Konzepte. Jedes Bewertungskriterium wird zunächst erläutert und anschließend bezüglich dessen Relevanz für die BSSD-Integration eingeschätzt. Die Einschätzung bezüglich der Relevanz der jeweiligen Kriterien basiert auf einer Diskussion zwischen den beiden Entwicklern der BSSD und dem Verfasser dieser Arbeit.

---

## 1. Zusätzliche Datenmenge

Dieses Bewertungskriterium betrachtet, wie viele zusätzliche Daten in die bestehende OpenDRIVE-Datei durch die Integration der BSSD-Struktur entsprechend des jeweiligen Konzepts eingefügt werden. Die zusätzliche Datenmenge ist beispielsweise über die Dateigröße der OpenDRIVE-Datei bestimmbar. Da eine OpenDRIVE-Datei einer einfachen Textdatei entspricht, wächst die Dateigröße beim Einfügen eines zusätzlichen Zeichens in der Regel<sup>63</sup> um ein Byte.

Das Bewertungskriterium der zusätzlichen Datenmenge weist eine eher niedrige Relevanz auf, da keine Einschränkungen durch die Dateigröße vorhandener OpenDRIVE-Dateien bekannt sind. Beispielsweise eine OpenDRIVE-Karte der Mittelstadt Griesheim, die repräsentative urbane Szenarien enthält, weist eine Dateigröße von ca. 8 MB auf. Bei der in der heutigen Zeit verfügbaren Rechenleistung ist davon auszugehen, dass eine zusätzliche Datenmenge im Bereich von einigen Megabyte keinen relevanten Einfluss auf die Nutzbarkeit der OpenDRIVE-Datei haben wird.

## 2. Rechenaufwand für die Positionszuordnung des BSSD-Verhaltensraums

Dieses Bewertungskriterium betrachtet den Rechenaufwand bzw. die Komplexität des Vorgangs, der erforderlich ist, um die BSSD-Verhaltensattribute an einer definierten Position der OpenDRIVE-Karte zu bestimmen. Beispielsweise wäre es bei der Verwendung einer mit der BSSD integrierten OpenDRIVE-Karte in einem Fahrzeug erforderlich, diesen Vorgang kontinuierlich durchzuführen. Weiterhin ist eine Visualisierung der mit der BSSD integrierten OpenDRIVE-Karte ein denkbarer Anwendungsfall, bei dem eine kontinuierliche Zuordnung der BSSD-Informationen zu einer bestimmten Position der OpenDRIVE-Karte erforderlich ist.

Diese beiden beispielhaften Anwendungsfälle zeigen, dass davon auszugehen ist, dass die Zuordnung des Verhaltensraums zu einer definierten Position der OpenDRIVE-Karte ein häufig erforderlicher Vorgang ist. Folglich weist das Bewertungskriterium des Rechenaufwands für die Positionszuordnung des BSSD-Verhaltensraums eine hohe Relevanz auf.

## 3. Beeinflussung der Struktur der OpenDRIVE-Streckendefinition

Bei diesem Bewertungskriterium wird überprüft, ob die vorhandene Struktur der Streckendefinition der OpenDRIVE-Karte durch das jeweilige Konzept beeinflusst wird. Die Struktur der OpenDRIVE-Streckendefinition umfasst alle Elemente der OpenDRIVE-Karte, die eine strukturelle Einteilung der Karte vornehmen, jedoch keinen Einfluss auf die Geometrie der Karte haben. Beispielsweise *lane sections* gehören zur Struktur der OpenDRIVE-Streckendefinition, da diese eine longitudinale Strukturierung der Straße vornehmen. Die OpenDRIVE-Fahrstreifen hingegen gehören nicht zur Struktur der OpenDRIVE-Streckendefinition, da sie die Geometrie der einzelnen Fahrstreifen und somit die Geometrie der gesamten Straße definieren.

Da OpenDRIVE ein offenes und weitverbreitetes Kartenformat ist, existieren verschiedenste Anwendungsfälle von OpenDRIVE-Karten mit unterschiedlichen Anforderungen an die OpenDRIVE-Karte. Somit ist schwierig abzuschätzen, ob die Nutzbarkeit der OpenDRIVE-Karte durch die Änderung der

---

<sup>63</sup> Bei allen Zeichen, die im ASCII-Zeichensatz enthalten sind

---

Struktur der Streckendefinition beeinflusst wird. Aufgrund dieser Einschätzung ist die Beeinflussung der Struktur der OpenDRIVE-Streckendefinition durch ein Konzept als ein relevanter Aspekt einzuschätzen.

#### **4. Aufwand zum Einfügen der BSSD-Struktur**

Dieses Bewertungskriterium betrachtet den notwendigen Aufwand, um die im jeweiligen Konzept definierte BSSD-Struktur in die OpenDRIVE-Karte zu integrieren. Dies beinhaltet zwei Aspekte. Einerseits den Aufwand für manuell durchzuführende Schritte, wie beispielsweise das Festlegen von Beginn und Ende einzelner BSSD-Segmente. Andererseits wird der Rechenaufwand für eine Durchführung der zu den jeweiligen Konzepten zugehörigen Algorithmen betrachtet.

Der für das Einfügen der BSSD-Struktur benötigte Aufwand ist bei jeder Integration der BSSD in eine neue OpenDRIVE-Karte erforderlich. Aufgrund dessen ist der Aufwand der dafür erforderlichen manuell durchzuführenden Schritte als relevant einzuschätzen. Der für die Integration der BSSD notwendige Rechenaufwand hingegen ist als weniger relevant einzustufen, da damit keine manuell durchzuführenden Schritte verknüpft sind. Somit ergibt sich insgesamt eine mittlere Relevanz des Aufwands zum Einfügen der BSSD-Struktur.

#### **5. Aufwand zum Entfernen der BSSD-Struktur**

Bei diesem Bewertungskriterium wird der erforderliche Aufwand betrachtet, um die in die OpenDRIVE-Datei entsprechend des jeweiligen Konzepts integrierte BSSD-Struktur wieder aus der OpenDRIVE-Datei zu entfernen. Das Ziel ist somit die OpenDRIVE-Datei auf den Zustand vor der BSSD-Integration zu setzen.

Es ist nicht davon auszugehen, dass das Entfernen der BSSD-Struktur aus der OpenDRIVE-Karte ein häufiger Anwendungsfall ist. Des Weiteren liegt die originale OpenDRIVE-Karte im Regelfall nach wie vor als separate OpenDRIVE-Datei vor. Aufgrund dessen ist der Aufwand zum Entfernen der BSSD-Struktur als weniger relevant einzuschätzen.

#### **6. Explizitheit des BSSD-Verhaltensraums**

Bei diesem Bewertungskriterium wird für das jeweilige Konzept überprüft, ob ein atomarer Verhaltensraum der BSSD explizit als ein solcher in der mit der BSSD-Struktur integrierten OpenDRIVE-Karte darstellbar ist. Beispielsweise bei Konzept 1, <lane>-Ebene, ist aufgrund der bereits in der OpenDRIVE-Karte vorhandenen *lane sections* denkbar, dass innerhalb von zwei aufeinanderfolgenden *lane sections* zwei identische BSSD-Verhaltensräume existieren, die zusammen einen atomaren Verhaltensraum der BSSD repräsentieren.

Eine explizite Repräsentation der BSSD-Verhaltensräume in der mit der BSSD-Struktur integrierten OpenDRIVE-Karte ist anzustreben. Jedoch ist davon auszugehen, dass eine Aufteilung eines Verhaltensraums in mehrere identische Sub-Verhaltensräume zu keiner Einschränkung der Nutzbarkeit der mit der BSSD integrierten OpenDRIVE-Karte führt. Aufgrund dessen ergibt sich für das Bewertungskriterium der Explizitheit des BSSD-Verhaltensraums eine mittlere Relevanz.

---

## 5.2.2 Gewichtung der Bewertungskriterien

Basierend auf der in Abschnitt 5.2.1 vorgenommenen Einschätzungen der Relevanz der Bewertungskriterien werden die Bewertungskriterien im Folgenden quantitativ gewichtet. Das Ziel dieser Gewichtung liegt in der Priorisierung von relevanten Kriterien im Vergleich zu weniger relevanten Kriterien. Die Gewichtung der Bewertungskriterien erfolgt anhand der Methode des Paarvergleichs. Dabei wird jedes mögliche Paar von Bewertungskriterien separat betrachtet. Für jedes dieser Paare wird festgelegt, welches der beiden Bewertungskriterien die höhere Relevanz aufweist bzw., dass beide Kriterien die gleiche Relevanz aufweisen. Das Bewertungskriterium mit der höheren Relevanz erhält zwei Punkte, während das Bewertungskriterium mit der niedrigeren Relevanz keinen Punkt erhält. Im Falle der gleichen Relevanz erhalten beide Kriterien einen Punkt. Aus der Durchführung des Paarvergleichs für alle möglichen Kombinationen der Bewertungskriterien lässt sich eine Gesamtsumme für jedes Bewertungskriterium errechnen. Die Gesamtsummen der Bewertungskriterien werden anschließend auf einen Wert von Eins normiert, sodass eine relative Gewichtung der Bewertungskriterien in Prozent resultiert.<sup>64</sup>

Die praktische Durchführung des Paarvergleichs erfolgt anhand einer Kreuztabelle, die für jedes Bewertungskriterium eine Zeile und eine Spalte enthält. Mit Ausnahme der Diagonalzellen, stellt jede Zelle eine Kombination von zwei Bewertungskriterien dar. Dabei existieren für jede Kombination von zwei Bewertungskriterien zwei Zellen. In diese beiden Zellen werden jeweils die Punkte aus Sicht des in der Zeile der Kreuztabelle befindlichen Bewertungskriteriums eingetragen. Aus der Summe der innerhalb einer Zeile befindlichen Punkte wird die Gewichtung der einzelnen Bewertungskriterien berechnet. Die Anwendung der Methode des Paarvergleichs auf die in Abschnitt 5.2.1 definierten Bewertungskriterien basierend auf der Einschätzung des Verfassers dieser Arbeit ist in Tabelle 5-1 dargestellt. Dabei repräsentiert das Symbol „+“ zwei Punkte, das Symbol „=“ einen Punkt und das Symbol „-“ keinen Punkt. Die aus dem Paarvergleich resultierenden absoluten sowie normierten Gewichtungen der Bewertungskriterien sind in den beiden rechten Spalten der Tabelle enthalten.

Da die Gewichtung der einzelnen Bewertungskriterien subjektiv ist, wird die Gewichtung der Bewertungskriterien zusätzlich von den beiden Entwicklern der BSSD, analog zu Tabelle 5-1, vorgenommen (siehe Anhang A.2, Tabelle A-2 und Tabelle A-3). Daraus resultieren drei verschiedene Gewichtungen der Bewertungskriterien. Die finalen Gewichtungen der Bewertungskriterien werden als Durchschnitt aus diesen drei Gewichtungen errechnet. Daraus resultiert die in Tabelle 5-2 gelistete finale Gewichtung der Bewertungskriterien. Es zeigt sich, dass der Rechenaufwand für die Positionszuordnung des BSSD-Verhaltensraums das mit Abstand relevanteste Bewertungskriterium ist. Es folgen die Bewertungskriterien „Beeinflussung der OpenDRIVE-Streckendefinition“, „Aufwand zum Einfügen der BSSD-Struktur“ und „Explizitheit des BSSD-Verhaltensraums“, die eine mittlere Relevanz aufweisen. Die Bewertungskriterien „zusätzliche Datenmenge“ und „Aufwand zum Entfernen der BSSD-Struktur“ weisen eine niedrige Relevanz auf.

---

<sup>64</sup> Kühnapfel, J. B.: Nutzwertanalysen in Marketing und Vertrieb (2014), S. 14–16.

Tabelle 5-1: Paarvergleich der Bewertungskriterien

	Zusätzliche Datenmenge	Rechenaufwand für die Positionszuordnung des BSSD-Verhaltensraums	Beeinflussung der Struktur der OpenDRIVE-Streckendefinition	Aufwand zum Einfügen der BSSD-Struktur	Aufwand zum Entfernen der BSSD-Struktur	Explizitheit des BSSD-Verhaltensraums	Summe	Normiert
1	Zusätzliche Datenmenge	-	-	-	=	-	1	0,03
2	Rechenaufwand für die Positionszuordnung des BSSD-Verhaltensraums	+	+	+	+	+	10	0,33
3	Beeinflussung der Struktur der OpenDRIVE-Streckendefinition	+	-	-	+	=	5	0,17
4	Aufwand zum Einfügen der BSSD-Struktur	+	+	-	+	+	8	0,27
5	Aufwand zum Entfernen der BSSD-Struktur	=	-	-	-	-	1	0,03
6	Explizitheit des BSSD-Verhaltensraums	+	-	-	+	=	5	0,17
							30	1

Tabelle 5-2: Finale Gewichtung der Bewertungskriterien

	Bewertungskriterium	Gewichtung
1	Zusätzliche Datenmenge	8 %
2	Rechenaufwand für die Positionszuordnung des BSSD-Verhaltensraums	33 %
3	Beeinflussung der Struktur der OpenDRIVE-Streckendefinition	21 %
4	Aufwand zum Einfügen der BSSD-Struktur	18 %
5	Aufwand zum Entfernen der BSSD-Struktur	1 %
6	Explizitheit des BSSD-Verhaltensraums	19 %
		100 %

### 5.2.3 Anwendung der Bewertungskriterien

Die drei in Abschnitt 5.1 abgeleiteten Konzepte werden im Folgenden auf die in Abschnitt 5.2.1 definierten Bewertungskriterien angewendet. Dazu werden die drei Konzepte für jedes Bewertungskriterium in eine Rangliste mit den Plätzen Eins, Zwei und Drei, entsprechend der Erfüllung des jeweiligen Bewertungskriteriums relativ zu den anderen beiden Konzepten, eingeordnet. Dabei repräsentiert eine niedrigere Platzierung eine bessere Erfüllung des jeweiligen Bewertungskriteriums. Falls mehrere Konzepte bezüglich eines Bewertungskriteriums eine gleiche Erfüllung aufweisen, werden einzelne Plätze der Rangliste mehrfach vergeben.

#### 1. Zusätzliche Datenmenge

Die aus einem Konzept resultierende zusätzliche Datenmenge ergibt sich nach Abschnitt 5.2.1 durch die Anzahl der Zeichen, die der OpenDRIVE-Datei durch die Realisierung des jeweiligen Konzepts hinzugefügt werden. Daraus folgt, dass die zusätzliche Datenmenge mit steigender Anzahl der hinzugefügten XML-Elemente zunimmt. Wie Abschnitt 5.1 zu entnehmen ist, benötigen die Konzepte umso mehr zusätzliche XML-Elemente, je weniger vorhandene Strukturen der OpenDRIVE-Karte genutzt werden. Folglich weist Konzept 1 die niedrigste zusätzliche Datenmenge auf, da bei diesem

---

Konzept im Vergleich zu den anderen beiden Konzepten die meisten existierenden Strukturen der OpenDRIVE-Karte übernommen werden. Es wird lediglich der Verhaltensraum in die OpenDRIVE-Fahrstreifen eingefügt. Konzept 2 fügt die BSSD-Struktur in die in der OpenDRIVE-Karte vorhandenen Straßen ein, wodurch mehr zusätzliche XML-Elemente als bei Konzept 1 resultieren. Bei Konzept 3 wird eine BSSD-Struktur komplett unabhängig von den vorhandenen Straßen der OpenDRIVE-Karte definiert. Aufgrund dessen werden bei Konzept 3 die meisten zusätzlichen XML-Elemente erzeugt. Es ergibt sich die folgende Rangliste bezüglich der Erfüllung des Bewertungskriteriums der zusätzlichen Datenmenge:

Konzept 1, <lane>-Ebene: Platz 1

Konzept 2, <road>-Ebene: Platz 2

Konzept 3, <OpenDRIVE>-Ebene: Platz 3

## 2. Rechenaufwand für die Positionszuordnung des BSSD-Verhaltensraums

Der aus einem Konzept resultierende Rechenaufwand für die Bestimmung des Verhaltensraums an einer definierten Position der OpenDRIVE-Karte ergibt sich daraus, wie komplex die Zuordnung des bei der Position der OpenDRIVE-Karte gültigen OpenDRIVE-<lane>-Elements zu dem Element ist, das den an dieser Stelle gültigen atomaren Verhaltensraum repräsentiert. Bei Konzept 1 ist ein atomarer Verhaltensraum direkt in einem OpenDRIVE-<lane>-Element enthalten. Aufgrund dessen ist die Zuordnung des Verhaltensraums zu einer Position der OpenDRIVE-Karte direkt und mit geringfügigem Rechenaufwand möglich. Bei Konzept 2 und Konzept 3 existieren separate BSSD-<lane>-Elemente, welche die atomaren Verhaltensräume beinhalten. Um bei diesen beiden Konzepten die beschriebene Zuordnung zu realisieren, ist demnach erforderlich, das zu dieser Position zugehörige BSSD-<lane>-Element zu finden. Dieser Vorgang weist bei Konzept 2 eine geringere Komplexität als bei Konzept 3 auf, da bei Konzept 2 nur eine Zuordnung innerhalb einer Straße von OpenDRIVE notwendig ist. Bei Konzept 3 ist zusätzlich das Finden des zu dieser Position der OpenDRIVE-Straße zugehörigen <way>-Elements erforderlich. Es ergibt sich die folgende Rangliste bezüglich der Erfüllung des Bewertungskriteriums des Rechenaufwands für die Positionszuordnung des BSSD-Verhaltensraums:

Konzept 1, <lane>-Ebene: Platz 1

Konzept 2, <road>-Ebene: Platz 2

Konzept 3, <OpenDRIVE>-Ebene: Platz 3

## 3. Beeinflussung der Struktur der OpenDRIVE-Streckendefinition

Die Anwendung des Bewertungskriterium der Beeinflussung der Struktur OpenDRIVE-Streckendefinition auf die einzelnen Konzepte resultiert in einer binäre Fallunterscheidung, da ein Konzept entweder die Struktur der OpenDRIVE-Streckendefinition beeinflusst oder dies nicht tut. Bei Konzept 1 ist das Auftrennen bzw. das Zusammenführen von vorhandenen *lane sections* der OpenDRIVE-Karte erforderlich (siehe Abschnitt 5.1.1). Folglich wird die Struktur der vorhandenen OpenDRIVE-Streckendefinition bei Konzept 1 verändert. Bei den Konzepten 2 und 3 hingegen werden die BSSD-Strukturen ausschließlich über das <userData>-Element in die OpenDRIVE-Karte eingefügt. Da das <userData>-Element keinen Einfluss auf die Struktur der OpenDRIVE-Streckendefinition hat,

---

wird diese bei den Konzepten 2 und 3 nicht beeinflusst. Es ergibt sich die folgende Rangliste bezüglich der Erfüllung des Bewertungskriteriums der Beeinflussung der Struktur der OpenDRIVE-Streckendefinition:

Konzept 1, <lane>-Ebene: Platz 3

Konzept 2, <road>-Ebene: Platz 1

Konzept 3, <OpenDRIVE>-Ebene: Platz 1

#### **4. Aufwand zum Einfügen der BSSD-Struktur**

Die Bewertung des Aufwands für das Einfügen der BSSD-Strukturen in die OpenDRIVE-Karte bei den einzelnen Konzepten ist ohne eine Realisierung dieser nur begrenzt möglich. Aufgrund dessen handelt es sich bei der Anwendung dieses Kriteriums nur um Abschätzungen, die auf einem Wissensstand vor der Implementierung beruhen.

In Bezug auf den ersten Aspekt dieses Bewertungskriteriums, den Aufwand der manuell durchzuführenden Schritte für das Einfügen der BSSD-Struktur, ist es bei jedem der drei Konzepte erforderlich, dass der Beginn und das Ende der BSSD-Segmente manuell definiert werden. Die Ursache dafür ist, dass in der OpenDRIVE-Karte nur begrenzt Informationen über die BSSD-Verhaltensattribute, deren Zustand den Beginn und das Ende der BSSD-Segmente definieren, vorhanden sind. Bei Konzept 3 ist zusätzlich erforderlich, dass Beginn und Ende der <way>-Elemente manuell definiert werden, da im Gegensatz zu den Konzepten 1 und 2 die vorhandenen Straßen der OpenDRIVE-Karte nicht verwendet werden. Somit ist davon auszugehen, dass der Aufwand der manuell durchzuführenden Schritte bei Konzept 3 höher als bei den Konzepten 1 und 2 ist.

In Bezug auf den zweiten Aspekt dieses Bewertungskriteriums, den Rechenaufwand der für die Durchführung der jeweiligen Konzepte benötigten Algorithmen, ist eine Beurteilung zum Wissensstand vor der Implementierung nicht realistisch durchführbar. Es ist lediglich vorhersehbar, dass für die Durchführung der von Konzept 3 benötigten Algorithmen ein höherer Rechenaufwand im Vergleich zur Durchführung der von Konzept 2 benötigten Algorithmen notwendig ist. Die Ursache dafür liegt darin, dass bei Konzept 3, äquivalent zu Konzept 2, die Zuordnung der BSSD-<lane>-Elemente zu den OpenDRIVE-<lane>-Elementen notwendig ist, jedoch zusätzlich eine Zuordnung der <way>-Elemente der BSSD-Struktur zu den <road>-Elementen der OpenDRIVE-Struktur erforderlich ist. Basierend auf den Abschätzungen bezüglich der Erfüllung des Bewertungskriteriums des Aufwands zum Einfügen der BSSD-Struktur resultiert die folgende Rangliste:

Konzept 1, <lane>-Ebene: Platz 1

Konzept 2, <road>-Ebene: Platz 1

Konzept 3, <OpenDRIVE>-Ebene: Platz 3

#### **5. Aufwand zum Entfernen der BSSD-Struktur**

Der aus einem Konzept resultierende Aufwand zum Entfernen der BSSD-Struktur ergibt sich aus den Schritten, die erforderlich sind, um die OpenDRIVE-Karte in den Zustand vor der BSSD-Integration zu versetzen. Mit Ausnahme von Konzept 1 beschränken sich die Modifikationen der OpenDRIVE-Datei durch die Umsetzung der Konzepte auf das Einfügen des Elements <userData> an

---

unterschiedlichen Stellen der Datei. Daraus ergibt sich, dass der Aufwand für das Entfernen der BSSD-Strukturen höher ist, je mehr `<userData>`-Elemente in die OpenDRIVE-Datei eingefügt werden. Nach einer Umsetzung von Konzept 3 existiert nur ein übergeordnetes `<userData>`-Element in der OpenDRIVE-Datei (siehe Abbildung 5-6). Bei Konzept 2 bzw. Konzept 1 existiert ein `<userData>`-Element je Straße (siehe Abbildung 5-4) bzw. OpenDRIVE-Fahstreifen (siehe Abbildung 5-2). Daher ist der Aufwand für das Entfernen der BSSD-Strukturen bei Konzept 3 am geringsten, gefolgt von Konzept 2, das wiederum von Konzept 1 gefolgt wird. Bei Konzept 1 ist zusätzlich erforderlich, dass das Auftrennen bzw. das Zusammenführen der *lane sections* rückgängig gemacht wird. Es ergibt sich die folgende Rangliste bezüglich der Erfüllung des Bewertungskriteriums des Aufwands zum Entfernen der BSSD-Struktur:

Konzept 1, `<lane>`-Ebene: Platz 3

Konzept 2, `<road>`-Ebene: Platz 2

Konzept 3, `<OpenDRIVE>`-Ebene: Platz 1

## 6. Explizitheit des BSSD-Verhaltensraums

Die Erfüllung des Bewertungskriteriums der Explizitheit des BSSD-Verhaltensraums bei den jeweiligen Konzepten hängt von der Struktur ab, welche die atomaren Verhaltensräume der BSSD beinhaltet. Bei Konzept 1 ist ein atomarer Verhaltensraum in einem OpenDRIVE-`<lane>`-Element enthalten, das longitudinal durch die *lane sections* der OpenDRIVE-Karte begrenzt ist. Folglich ist erforderlich, dass Beginn und Ende der *lane sections* entsprechend der Segmentierung der BSSD definiert sind, um eine explizite Repräsentation eines atomaren Verhaltensraums zu ermöglichen. Um dies zu realisieren ist das Auftrennen bzw. das Zusammenführen der vorhandenen *lane sections* erforderlich. Nach der Einschätzung des Verfassers dieser Arbeit scheint das Auftrennen vorhandener *lane sections* möglich, jedoch ein Zusammenführen vorhandener *lane sections* bei gleichzeitiger Beibehaltung der Funktionsfähigkeit der OpenDRIVE-Karte unrealistisch. Aufgrund dessen ist davon auszugehen, dass bei Konzept 1 der Fall auftritt, dass ein atomarer Verhaltensraum der BSSD auf mehrere *lane sections* aufgeteilt ist, wenn der Definitionsbereich dieses atomaren Verhaltensraums über den Bereich einer *lane section* hinausgeht (siehe beispielsweise Abbildung 5-5). Darüber hinaus ist es möglich, dass ein atomarer Verhaltensraum für einen Bereich definiert ist, der über den Definitionsbereich einer Straße der OpenDRIVE-Karte hinausgeht. Das Auftreten dieses Falls ist insbesondere möglich, weil seitens der Spezifikation von OpenDRIVE keine Vorgaben bezüglich des Definitionsbereichs einer Straße<sup>65</sup> existieren und dementsprechend ein Aneinanderreihen von mehreren kurzen Straßen möglich ist. In diesem Fall wäre eine explizite Repräsentation dieses atomaren Verhaltensraums ebenfalls nicht darstellbar, da die *lane sections* nur für eine bestimmte Straße der OpenDRIVE-Karte definiert sind. Bei Konzept 2 ist ein atomarer Verhaltensraum in einem BSSD-`<lane>`-Element enthalten, das longitudinal durch die durch das Konzept erstellten BSSD-Segmente begrenzt ist. Die BSSD-Segmente werden entsprechend der Begrenzungen der atomaren

---

<sup>65</sup> Eine beispielhafte Vorgabe wäre, dass Beginn und Ende eine Straße immer mit einer Kreuzung verbunden sein müssen (siehe beispielsweise `<way>`-Elemente der BSSD, Abschnitt 2.5.1), sofern der Beginn oder das Ende der Straße nicht ein Ende des Definitionsbereichs der OpenDRIVE-Karte repräsentieren

Verhaltensräume definiert. Aufgrund dessen ist eine explizite Repräsentation der atomaren Verhaltensräume bei Konzept 2 grundsätzlich möglich. Eine Ausnahme davon bildet der bereits bei Konzept 1 angesprochene Spezialfall, dass ein atomarer Verhaltensraum über den Definitionsbereich einer Straße der OpenDRIVE-Karte hinausgeht. Da die BSSD-Segmente bei Konzept 2 innerhalb einer Straße von OpenDRIVE definiert sind, wäre in diesem Fall keine explizite Repräsentation des atomaren Verhaltensraums möglich. Bei Konzept 3 ist ein atomarer Verhaltensraum in einem BSSD-`<lane>`-Element enthalten, das longitudinal durch die durch das Konzept erstellten BSSD-Segmente begrenzt ist. Analog zu Konzept 2 ist somit eine explizite Repräsentation der atomaren Verhaltensräume bei Konzept 3 möglich. Die BSSD-Segmente werden innerhalb der `<way>`-Elemente der BSSD-Struktur definiert, deren Begrenzungen unabhängig von den bereits vorhandenen OpenDRIVE-Straßen festlegbar sind. Aufgrund dessen erscheint auch eine explizite Repräsentation eines atomaren Verhaltensraums, der über die Grenzen einer Straße der OpenDRIVE-Karte definiert ist, möglich. Es ergibt sich die folgende Rangliste bezüglich der Erfüllung des Bewertungskriteriums der Explizitheit des BSSD-Verhaltensraums:

- Konzept 1, `<lane>`-Ebene: Platz 3
- Konzept 2, `<road>`-Ebene: Platz 2
- Konzept 3, `<OpenDRIVE>`-Ebene: Platz 1

Die Anwendung der Bewertungskriterien auf die abgeleiteten Konzepte ist zusammenfassend in Tabelle 5-3 gelistet.

Tabelle 5-3: Anwendung der Bewertungskriterien auf abgeleitete Konzepte

	<b>Bewertungskriterium</b>	<b>Konzept 1</b> <code>&lt;lane&gt;</code> - Ebene	<b>Konzept 2</b> <code>&lt;road&gt;</code> - Ebene	<b>Konzept 3</b> <code>&lt;OpenDRIVE&gt;</code> - Ebene
1	Zusätzliche Datenmenge	1	2	3
2	Rechenaufwand für die Positionszuordnung des BSSD-Verhaltensraums	1	2	3
3	Beeinflussung der Struktur der OpenDRIVE-Streckendefinition	3	1	1
4	Aufwand zum Einfügen der BSSD-Struktur	1	1	3
5	Aufwand zum Entfernen der BSSD-Struktur	3	2	1
6	Explizitheit des BSSD-Verhaltensraums	3	2	1

#### 5.2.4 Nutzwertanalyse

Auf Basis der Gewichtung der Bewertungskriterien (siehe Tabelle 5-2) und der Anwendung dieser auf die einzelnen Konzepte (siehe Tabelle 5-3) ist die Durchführung einer Nutzwertanalyse zur Auswahl des Konzepts mit dem höchsten Gesamtnutzen möglich. Dazu wird für jede Kombination eines Konzepts und Bewertungskriteriums ein Teilnutzwert berechnet. Dieser Teilnutzwert wird durch die Multiplikation der Gewichtung des Bewertungskriteriums mit der Punktzahl der Erfüllung dieses Bewertungskriteriums durch das Konzept errechnet. Damit eine bessere Erfüllung eines

Bewertungskriteriums zu einem höheren Nutzwert führt, ist erforderlich, dass die Zahlenwerte der in Abschnitt 5.2.3 angewendete Rangliste umgekehrt werden. Folglich resultieren für die Plätze Eins, Zwei und Drei die Punktzahlen Drei, Zwei und Eins. Bei Bewertungskriterien, deren Anwendung auf die Konzepte in einer doppelten Vergabe einer Platzierung resultierte, wird die Punktzahl aus dem Durchschnitt der durch die beiden in Anspruch genommenen Platzierungen errechnet. Durch die Summierung der zu einem Konzept zugehörigen Teilnutzwerte wird ein Gesamtnutzwert für jedes Konzept berechnet. Die resultierende Nutzwertanalyse ist in Tabelle 5-4 dargestellt. Es ist zu erkennen, dass Konzept 2 den höchsten Gesamtnutzwert, Konzept 1 den zweithöchsten Gesamtnutzwert und Konzept 3 den niedrigsten Gesamtnutzwert aufweist.

Tabelle 5-4: Nutzwertanalyse zur Auswahl eines Konzepts

			Konzept 1		Konzept 2		Konzept 3	
			<lane>-Ebene		<road>-Ebene		<OpenDRIVE>-Ebene	
	Bewertungskriterium	Gewichtung in %	Bewertung	Nutzwert	Bewertung	Nutzwert	Bewertung	Nutzwert
1	Zusätzliche Datenmenge	8	3	24	2	16	1	8
2	Rechenaufwand für die Positionszuordnung des BSSD-Verhaltensraums	33	3	99	2	66	1	33
3	Beeinflussung der Struktur der OpenDRIVE-Streckendefinition	21	1	21	2,5	52,5	2,5	52,5
4	Aufwand zum Einfügen der BSSD-Struktur	18	2,5	45	2,5	45	1	18
5	Aufwand zum Entfernen der BSSD-Struktur	1	1	1	2	2	3	3
6	Explizitheit des BSSD-Verhaltensraums	19	1	19	2	38	3	57
<b>Summe</b>		<b>100%</b>		<b>209</b>		<b>219,5</b>		<b>171,5</b>

Um das Risiko einer Fehlentscheidung durch eine Nutzwertanalyse zu verringern, empfiehlt Kühnapfel<sup>66</sup> im Anschluss die Sensibilität dieser zu überprüfen. Dabei wird die Robustheit des Ergebnisses der Nutzwertanalyse durch Variation der beiden Stellgrößen (Gewichtung der Bewertungskriterien, Bewertungsschema für die Anwendung der Bewertungskriterien) geprüft. Für die Variation der Gewichtungen wird die Nutzwertanalyse in Tabelle 5-4 zusätzlich jeweils mit den Gewichtungen des Verfassers dieser Arbeit (siehe Tabelle 5-1) sowie den Gewichtungen der beiden Entwickler der BSSD (siehe Tabelle A-2 und Tabelle A-3) durchgeführt. In allen drei Fällen resultiert die gleiche Rangordnung des Gesamtnutzwerts der Konzepte wie in Tabelle 5-4. Für die Variation des Bewertungsschemas erfolgt die Anwendung der Bewertungskriterien auf die Konzepte anhand einer Schulnotenskala (Punktzahlen 6–1 entsprechend der Schulnoten 1–6), die im Vergleich zum relativen Bewertungsschema in Abschnitt 5.2.3 eine absolute Bewertung repräsentiert. Auch bei dieser Variation resultiert die gleiche Rangordnung des Gesamtnutzwerts der Konzepte wie in Tabelle 5-4 (siehe Tabelle A-4).

<sup>66</sup> Kühnapfel, J. B.: Nutzwertanalysen in Marketing und Vertrieb (2014), S. 19–20.

---

Es zeigt sich, dass das Ergebnis der Nutzwertanalyse eine Robustheit gegenüber Schwankungen der Gewichtungen sowie des Bewertungsschemas aufweist. Folglich ist davon auszugehen, dass eine Umsetzung von Konzept 2 tatsächlich den höchsten Nutzen bietet. Eine mögliche Ursache dafür ist, dass Konzept 2 in Bezug auf alle Bewertungskriterien einen guten Mittelweg darstellt. Bei der Anwendung der Bewertungskriterien wurde Konzept 2 stets Platz Zwei oder ein geteilter Platz Eins zugewiesen (siehe Tabelle 5-3). Insbesondere ein mittlerer Rechenaufwand bei der Positionszuordnung des Verhaltensraum, keine Beeinflussung der OpenDRIVE-Streckendefinition sowie eine meist explizite Repräsentation des Verhaltensraums sind Hauptargumente für die Umsetzung von Konzept 2. Losgelöst von den Bewertungskriterien erscheint Konzept 2 im Vergleich zu den anderen beiden Konzepten am geeignetsten, da bei diesem Konzept die der BSSD-Spezifikation ähnlichen Elemente von OpenDRIVE, `<road>` und `<junction>`, übernommen werden und die BSSD-Struktur bei der Stufe der OpenDRIVE-Hierarchie, bei der Unterschiede zwischen der BSSD und OpenDRIVE bestehen – der longitudinalen Unterteilungen einer Straße in *lane sections*, welche die Geometrie der Straße definieren, sowie BSSD-Segmente, die den in dieser Straße gültigen Verhaltensraum definieren – eingefügt wird.

Resultierend aus dem Ergebnis der Nutzwertanalyse, der Überprüfung der Robustheit dieses Ergebnisses sowie der Betrachtung der inhaltlichen Ursachen für den höchsten Gesamtnutzen von Konzept 2, wird Konzept 2 als das Konzept ausgewählt, das in dieser Arbeit im Folgenden ausdetailliert, bezüglich einer Umsetzung bewertet und je nach Umsetzungsbewertung zu Teilen oder vollständig umgesetzt wird.

### 5.3 Detaillierung des ausgewählten Konzepts

In diesem Abschnitt wird das im vorigen Abschnitt ausgewählte Konzept (Konzept 2, `<road>`-Ebene) basierend auf dem in Abschnitt 5.1.2 abgeleiteten Grundprinzip ausdetailliert. Dazu werden einzelne zur Umsetzung von Konzept 2 notwendige Schritte definiert (Abschnitt 5.3.1) und bezüglich einer Umsetzung bewertet (Abschnitt 5.3.2).

Als Ausgangspunkt für die Ableitung der zur Umsetzung des Konzepts notwendigen Schritte zeigt Abbildung 5-7 die vollständige aus Konzept 2 resultierende Gesamtstruktur der BSSD-Integration in das OpenDRIVE-Kartenformat. Alle aus der BSSD-Integration resultierenden Elemente sind in Orange dargestellt. In eine Straße der OpenDRIVE-Karte wird unterhalb des je Straße genau einmal existierenden `<lanes>`-Element das Element `<userData>` eingefügt, welches das Einfügen benutzerdefinierter Daten ermöglicht. Dem `<userData>`-Element werden beliebig viele BSSD-Segmente untergeordnet, die jeweils durch ein `<segment>`-Element repräsentiert werden. Entsprechend der Spezifikation der BSSD verfügt ein BSSD-Segment über die untergeordneten Elemente `<right>` und `<left>`, die der Gruppierung der BSSD-`<lane>`-Elemente dienen. Letztere beinhalten einerseits die untergeordneten Elemente `<behaviorAlong>` und `<behaviorAgainst>`, die jeweils den in dem BSSD-`<lane>`-Element gültigen atomaren Verhaltensraum beinhalten. Andererseits ist erforderlich, dass jedes BSSD-`<lane>`-Element über eine untergeordnete Hilfsstruktur verfügt, die eine eindeutige Zuordnung dieses BSSD-`<lane>`-Elements zu einem OpenDRIVE-`<lane>`-Element

für jede  $s$ -Koordinate, bei der das übergeordnete BSSD-Segment definiert ist, ermöglicht. Die Definition einer Hilfsstruktur erfolgt im Rahmen der in diesem Abschnitt beschriebenen Detaillierung des Konzepts. Die einem BSSD-`<lane>`-Element vorangehenden und nachfolgenden BSSD-`<lane>`-Elemente sind implizit über die `<predecessor>` bzw. `<successor>`-Elemente des OpenDRIVE-`<lane>`-Elements, das mit der Hilfsstruktur zu diesem BSSD-`<lane>`-Element verknüpft ist, bekannt. Aufgrund dessen ist keine explizite Angabe des vorangehenden und nachfolgenden BSSD-`<lane>`-Elements erforderlich.

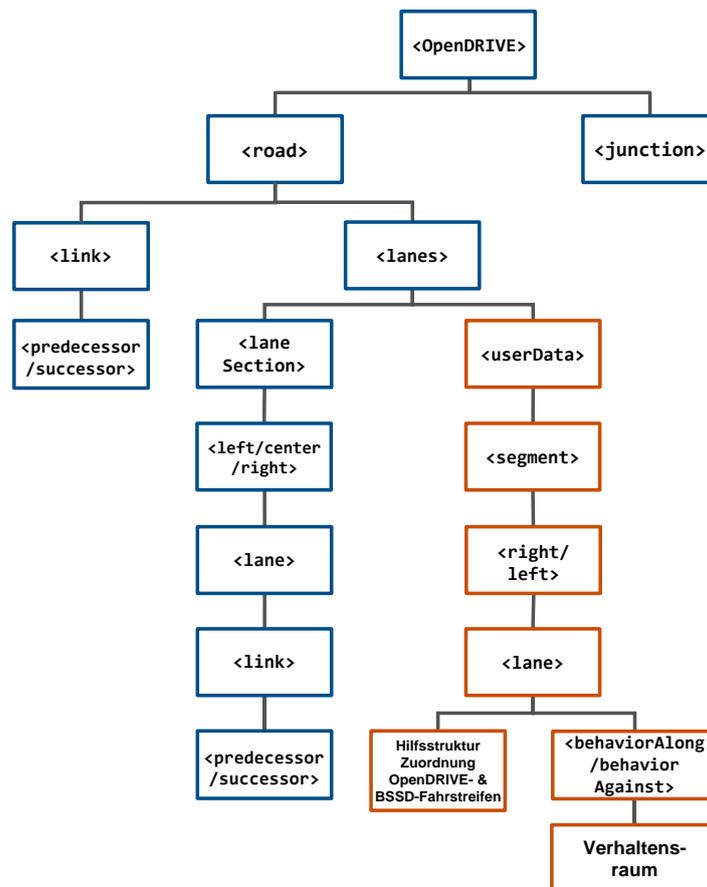


Abbildung 5-7: Gesamtstruktur der BSSD-Integration in OpenDRIVE

### 5.3.1 Ableitung von Schritten

Um die in Abbildung 5-7 dargestellte Struktur für eine bestehende OpenDRIVE-Datei zu realisieren, ist erforderlich, die aus der BSSD-Integration resultierenden Elemente sukzessive für alle Straßen der OpenDRIVE-Karte zu erstellen und einzufügen. Aus den einzelnen einzufügenden Elementen ergeben sich die zur Umsetzung von Konzept 2 notwendigen Schritte. Diese sind in Abbildung 5-8 dargestellt und werden im Folgenden näher erläutert. Es sei angemerkt, dass die beschriebenen Schritte ein ideales Vorgehen für die ganzheitliche Integration der BSSD in eine OpenDRIVE-Karte darstellen. Daher werden diese Schritte darüber hinaus bezüglich einer Umsetzung im Rahmen dieser Arbeit bewertet (siehe Abschnitt 5.3.2).

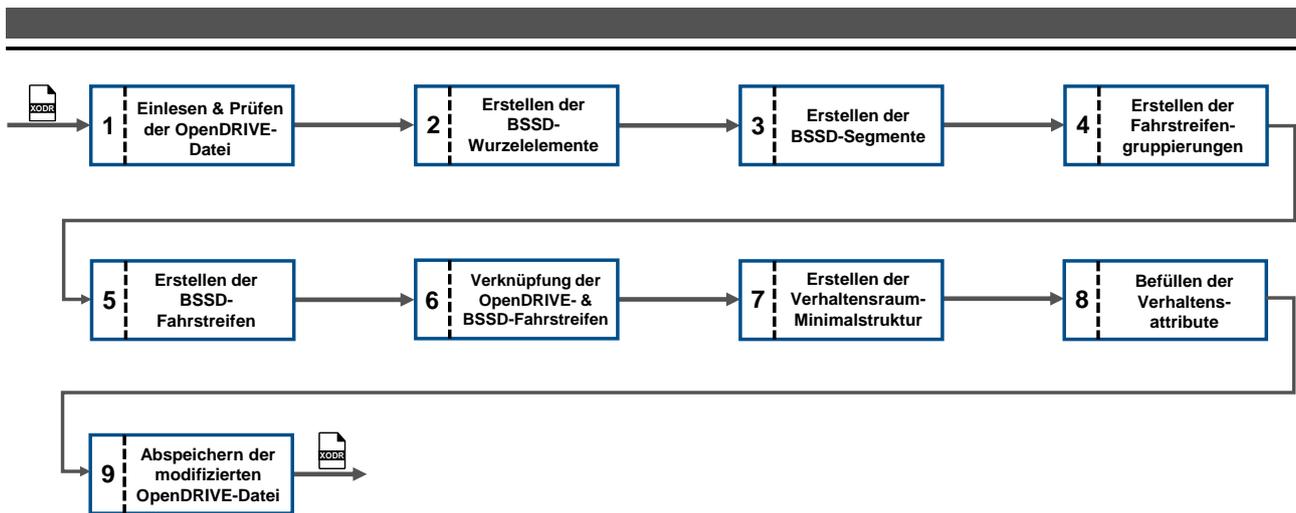


Abbildung 5-8: Schritte zur Umsetzung von Konzept 2

### 5.3.1.1 Schritt 1: Einlesen & Prüfen der OpenDRIVE-Datei

Um die Integration der BSSD in eine OpenDRIVE-Datei erfolgreich durchzuführen, ist vor der eigentlichen Verarbeitung der OpenDRIVE-Datei erforderlich, zu überprüfen, ob die OpenDRIVE-Datei den Vorgaben der offiziellen Spezifikation der entsprechenden Version von OpenDRIVE<sup>67,68</sup> entspricht. Da OpenDRIVE ein offener Standard ist, besteht die Möglichkeit, dass OpenDRIVE-Karten in Verwendung sind, die nicht in Übereinstimmung mit den Vorgaben der entsprechenden Spezifikation erstellt wurden. In diesem Fall ist es nicht möglich, die korrekte Durchführung aller Konzept-Schritte zu garantieren. Entspricht die OpenDRIVE-Datei den Vorgaben der Spezifikation wird diese als *valide* bezeichnet. Für die Überprüfung der Validität einer OpenDRIVE-Datei stehen für die einzelnen Versionsveröffentlichungen von OpenDRIVE XML-Schema-Dateien zur Verfügung (Dateiformat XSD). Diese beinhalten beispielsweise Vorgaben zu den in der Datei zu verwendenden Elementen, Attributen sowie Datentypen für die einzelnen Attribute. Ist die Validität der zu verwendenden OpenDRIVE-Datei bestätigt, sind die Rahmenbedingungen geschaffen, um eine korrekte und vollständige Integration der BSSD in die OpenDRIVE-Datei durchzuführen.

### 5.3.1.2 Schritt 2: Erstellen der BSSD-Wurzelemente

Entsprechend Abbildung 5-7 beginnt die BSSD-Integration in eine OpenDRIVE-Karte mit dem Erstellen des Elements `<userData>`. Da dieses Element das oberste Element der einzufügenden BSSD-Struktur darstellt, wird dieses Element auch als *BSSD-Wurzelement* bezeichnet. Das BSSD-Wurzelement wird grundsätzlich in jeder Straße der OpenDRIVE-Karte erstellt. Ausnahmen davon sind möglich, da sich die Modellierungsgrenzen von OpenDRIVE und BSSD unterscheiden. OpenDRIVE verfolgt das Ziel, ein vorhandenes Verkehrsnetzwerk so vollständig wie möglich zu modellieren. Dementsprechend modelliert OpenDRIVE neben der durch Kraftfahrzeuge verwendeten Fahrbahn auch Teile des Verkehrsnetzwerks, die sich neben der Fahrbahn befinden, wie beispielsweise Geh- oder Fahrradwege. Diese werden in OpenDRIVE analog zu den OpenDRIVE-Fahrstreifen der

<sup>67</sup> ASAM e.V.: ASAM OpenDRIVE® - Previous Releases (2021).

<sup>68</sup> ASAM e.V.: ASAM OpenDRIVE® (2021).

---

Fahrbahn als OpenDRIVE-`<lane>`-Elemente modelliert. Die Definition der BSSD hingegen ist auf die Fahrbahn beschränkt (siehe Abschnitt 2.4). Folglich ist es erforderlich, die BSSD-Informationen nur für alle OpenDRIVE-`<lane>`-Elemente, die einen Teil der Fahrbahn repräsentieren<sup>69</sup>, festzulegen. Für alle OpenDRIVE-`<lane>`-Elemente, die keinen Teil der Fahrbahn darstellen<sup>70</sup>, ist dementsprechend nicht erforderlich, BSSD-Informationen zu speichern. Daraus ergibt sich der Spezialfall, dass eine Straße der OpenDRIVE-Karte ausschließlich aus nicht befahrbaren OpenDRIVE-Fahrstreifen besteht (z.B. Fußgängerzone). In dieser Straße wäre es demnach nicht erforderlich, eine BSSD-Struktur einzufügen.

Es zeigt sich, dass das Erstellen des BSSD-Wurzelements nur in Straßen erforderlich ist, die mindestens einen befahrbaren OpenDRIVE-Fahrstreifen besitzen. Somit ist als Voraussetzung für das korrekte Erstellen der BSSD-Wurzelemente die Durchführung eines Algorithmus erforderlich, der alle vorhandenen OpenDRIVE-Fahrstreifen in die Kategorien *befahrbar* und *nicht-befahrbar* klassifiziert. Basierend auf dieser Einteilung wird in jeder Straße, die mindestens einen befahrbaren OpenDRIVE-Fahrstreifen besitzt, ein BSSD-Wurzelement erstellt.

### 5.3.1.3 Schritt 3: Erstellen der BSSD-Segmente

Im dritten Schritt der BSSD-Integration werden in jeder Straße, die mindestens einen befahrbaren OpenDRIVE-Fahrstreifen besitzt, die Segmente der BSSD erstellt. Dabei wird ein BSSD-Segment durch das Element `<segment>` repräsentiert. Ein BSSD-Segment ist ein in longitudinaler Richtung abgegrenzter Abschnitt einer Straße, der die in dem BSSD-Segment enthaltenen atomaren Verhaltensräume longitudinal begrenzt. Aufgrund dessen ist für die Definition und eindeutige Identifizierung eines BSSD-Segments erforderlich festzulegen, bei welcher *s*-Koordinate der Straße das Segment beginnt und bei welcher *s*-Koordinate das Segment endet. Dazu wird dem `<segment>`-Element das verpflichtende Attribut `sStart` zugewiesen, das, analog zum Attribut `s` des `<laneSection>`-Elements von OpenDRIVE, die *s*-Koordinate angibt, bei der das BSSD-Segment beginnt. Grundsätzlich ist die Angabe einer *s*-Koordinate für die Definition von Beginn und Ende eines BSSD-Segments ausreichend, da entsprechend der Konvention der *lane sections* festgelegt wird, dass ein BSSD-Segment ab der im Attribut `sStart` angegebenen *s*-Koordinate bis zum Beginn des nachfolgenden BSSD-Segments bzw. bis zum Ende der Straße, falls kein nachfolgendes BSSD-Segment existiert, definiert ist.

Aufgrund der Einteilung der OpenDRIVE-Fahrstreifen in befahrbare und nicht-befahrbare OpenDRIVE-Fahrstreifen (siehe Abschnitt 5.3.1.2) ist der Spezialfall möglich, dass innerhalb einer Straße zugleich *lane sections*, die befahrbare OpenDRIVE-Fahrstreifen besitzen, und *lane sections*, die keine befahrbaren OpenDRIVE-Fahrstreifen besitzen, existieren. Eine beispielhafte Szenerie dazu, die eine in einer Sackgasse endende Straße repräsentiert, ist in Abbildung 5-9 dargestellt. Die dargestellte Straße besteht aus *lane section A*, die zwei befahrbare OpenDRIVE-Fahrstreifen besitzt, sowie aus *lane section B*, die aus vier nicht-befahrbaren OpenDRIVE-Fahrstreifen besteht, die

---

<sup>69</sup> In dieser Arbeit im Folgenden als *befahrbare OpenDRIVE-Fahrstreifen* bezeichnet

<sup>70</sup> In dieser Arbeit im Folgenden als *nicht-befahrbare OpenDRIVE-Fahrstreifen* bezeichnet

beispielhaft einen ausschließlich für Fußgänger und Fahrradfahrer zugänglichen Bereich repräsentieren, der durch Poller von *lane section* A getrennt ist.

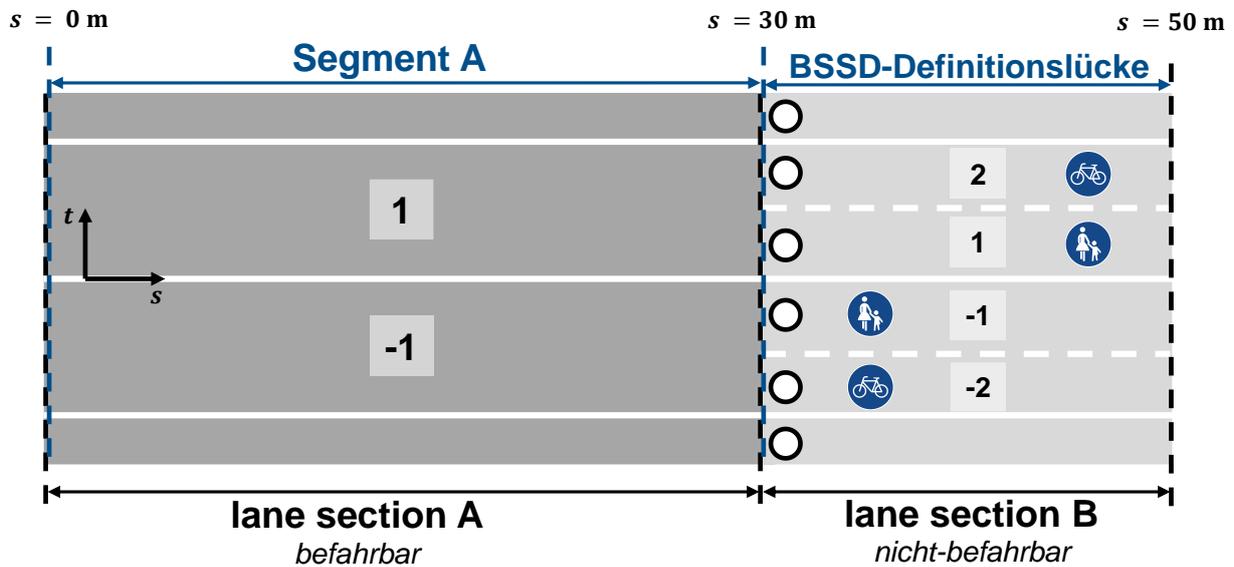


Abbildung 5-9: Beispiel Straße mit befahrbaren und nicht-befahrbaren OpenDRIVE-Fahrstreifen

In dem Bereich einer Straße innerhalb dessen eine *lane section* ohne befahrbare OpenDRIVE-Fahrstreifen existiert, ist eine Definition eines BSSD-Segments nicht zweckmäßig, da in diesem Bereich nur OpenDRIVE-Fahrstreifen vorhanden sind, die nicht mit der BSSD modelliert werden. Folglich ist es möglich, dass innerhalb bestimmter Bereiche einer Straße Definitionslücken der BSSD existieren. Eine BSSD-Definitions-lücke beschreibt somit einen Bereich einer Straße innerhalb dessen keine BSSD-Informationen vorhanden sind. Um die Angabe dieser Definitionslücken in der BSSD-Struktur zu ermöglichen, wird dem `<segment>`-Element das optionale Attribut `sEnd` zugewiesen. Dieses Attribut gibt, für den Fall, dass auf das BSSD-Segment eine BSSD-Definitions-lücke folgt, die `s`-Koordinate an, bei der das Segment endet. Die BSSD-Definitions-lücke ist somit ab der im Attribut `sEnd` angegebenen `s`-Koordinate bis zum Beginn eines nachfolgenden BSSD-Segments bzw. bis zum Ende der Straße, falls kein nachfolgendes BSSD-Segment existiert, definiert. In dem in Abbildung 5-9 dargestellten Beispiel existiert somit das Segment A mit den Attributen `sStart="0.0"` und `sEnd="30.0"`. Weiterhin existiert eine BSSD-Definitions-lücke, deren Definitionsbereich durch die Angabe von `sEnd="30.0"` und die Länge der Straße (`s = 50.0 m`) eindeutig festgelegt ist.

Zum Erstellen der BSSD-Segmente ist die Durchführung eines Algorithmus erforderlich, der anhand der in der OpenDRIVE-Karte vorhandenen BSSD-relevanten Informationen automatisiert die `s`-Koordinaten bestimmt, bei denen die Definition eines BSSD-Segments bzw. einer BSSD-Definitions-lücke erforderlich ist. Dieser Algorithmus basiert auf der Definition von Grundregeln, deren Erfüllung garantiert, dass bei einer bestimmten `s`-Koordinate die Definition des Beginns bzw. Endes eines BSSD-Segments erforderlich ist. Eine einfache beispielhafte Grundregel ist, dass beim Beginn der ersten *lane section* einer Straße, die befahrbare OpenDRIVE-Fahrstreifen beinhaltet, ein BSSD-Segment beginnt. Aus dieser Grundregel wäre beispielsweise in der in Abbildung 5-9 dargestellten Szenerie die Definition des Segments A mit dem Attribut `sStart="0.0"` ableitbar. Bei der Umsetzung des Konzepts werden weitere Grundregeln formuliert und in dem beschriebenen Algorithmus

---

zusammengefasst, um die BSSD-Segmente automatisiert aus der gegebenen OpenDRIVE-Karte zu extrahieren. Anhand Abbildung 5-10 ist es möglich, die Extraktion von BSSD-Segmenten anhand bestehender *lane sections* beispielhaft nachzuvollziehen.

#### 5.3.1.4 Schritt 4: Erstellen der Fahrstreifen-Gruppierungen

Im nächsten Schritt der BSSD-Integration werden unterhalb jedem erstellten `<segment>`-Element die beiden Elemente `<left>` und `<right>` eingefügt. Diese Elemente gruppieren die BSSD-Fahrstreifen in linke und rechte (Betrachtung der Mittellinie der Straße in positiver *s*-Richtung) BSSD-Fahrstreifen, um die Navigation eines menschlichen Lesers durch die Datei zu vereinfachen.

#### 5.3.1.5 Schritt 5: Erstellen der BSSD-Fahrstreifen

Im fünften Schritt der BSSD-Integration werden in jedem BSSD-Segment die BSSD-Fahrstreifen unterhalb der Elemente `<left>` bzw. `<right>` entsprechend ihrer zugehörigen Straßenseite erstellt. Dabei wird ein BSSD-Fahrstreifen durch ein BSSD-`<lane>`-Element repräsentiert. Zur Eindeutigen Identifikation eines BSSD-Fahrstreifens und der Bestimmung der lateralen Anordnung der BSSD-Fahrstreifen wird einem BSSD-Fahrstreifen entsprechend der Spezifikation der BSSD das verpflichtende Attribut `id` zugewiesen (siehe Abschnitt 2.5.1.2).

Ein BSSD-Fahrstreifen beinhaltet einen atomaren Verhaltensraum und ist in longitudinaler Richtung durch den Anfang und das Ende des übergeordneten BSSD-Segments begrenzt. Die laterale Begrenzung eines BSSD-Fahrstreifens basiert bei diesem Konzept auf den in der Karte vorhandenen OpenDRIVE-Fahrstreifen. Bei korrekter Definition der BSSD-Segmente bleibt die Anzahl der befahrbaren OpenDRIVE-Fahrstreifen, die in den *lane sections*, die mit diesem BSSD-Segment überschneiden<sup>71</sup>, enthalten sind, konstant. Dies bedeutet, dass innerhalb des Definitionsbereichs eines BSSD-Segments alle darin vorhandenen befahrbaren OpenDRIVE-Fahrstreifen einen eindeutigen vorangehenden bzw. nachfolgenden befahrbaren OpenDRIVE-Fahrstreifen besitzen, falls mehrere zu diesem BSSD-Segment überschneidende *lane sections* existieren. Somit ist eine Definition der BSSD-Fahrstreifen basierend auf den OpenDRIVE-Fahrstreifen möglich. Es wird die Regel aufgestellt, dass die BSSD-Fahrstreifen eines BSSD-Segments anhand der befahrbaren OpenDRIVE-Fahrstreifen der ersten *lane section*, die mit diesem BSSD-Segment überschneidet, definiert werden. Dies bedeutet, dass für jeden befahrbaren OpenDRIVE-Fahrstreifen, der in der ersten mit dem BSSD-Segment überschneidenden *lane section* enthalten ist, ein BSSD-Fahrstreifen mit der gleichen Ausprägung des `id`-Attributs erstellt wird. Falls nachfolgende *lane sections* existieren, die mit dem BSSD-Segment überschneiden, sind alle in diesen *lane sections* enthaltenen befahrbaren OpenDRIVE-Fahrstreifen eindeutig den befahrbaren OpenDRIVE-Fahrstreifen in der ersten mit dem BSSD-Segment überschneidenden *lane section* und somit auch den BSSD-Fahrstreifen zuzuordnen. Diese Zuordnung erfolgt über die Angabe der Vorgänger- bzw. Nachfolgerelemente (Elemente `<predecessor>` und `<successor>`).

---

<sup>71</sup> Eine Überschneidung einer *lane section* mit einem BSSD-Segment bedeutet in diesem Kontext, dass der Definitionsbereich der *s*-Koordinate der *lane section* mit dem Definitionsbereich der *s*-Koordinate des BSSD-Segments überschneidet

Zur Verdeutlichung der Erstellung der BSSD-Segmente und -Fahrstreifen auf Basis der vorhandenen OpenDRIVE-Fahrstreifen ist in Abbildung 5-10 eine beispielhafte Szenerie bestehend aus drei *lane sections* und zwei BSSD-Segmenten dargestellt. Analog zu den bisherigen Abbildungen sind die *id*-Attribute der OpenDRIVE- bzw. BSSD-Fahrstreifen in schwarz bzw. blau dargestellt. Zunächst werden die BSSD-Segmente basierend auf den vorhandenen *lane sections* abgeleitet. Am Beginn der Straße ( $s = 0$  m) wird *lane section* A definiert, die zwei befahrbare OpenDRIVE-Fahrstreifen besitzt (*id* 1 und -1). Da *lane section* A befahrbare OpenDRIVE-Fahrstreifen besitzt, wird an der identischen Stelle der Beginn eines BSSD-Segments (Segment A) definiert. Bei einer  $s$ -Koordinate von 20 m wird *lane section* B definiert, da zwei neue OpenDRIVE-Fahrstreifen entstehen. Einer dieser neuen OpenDRIVE-Fahrstreifen (*id* 2) ist befahrbar, während der andere neue OpenDRIVE-Fahrstreifen (*id* -1) einen nur für Fußgänger zugänglichen Bereich repräsentiert (z.B. Verkehrsinsel) und somit nicht-befahrbar ist. Da an dieser Stelle ein neuer befahrbarer OpenDRIVE-Fahrstreifen (*id* 2) definiert wird, ist ebenfalls die Definition eines neuen BSSD-Segments (Segment B) erforderlich. Bei einer  $s$ -Koordinate von 50 m teilt sich der OpenDRIVE-Fahrstreifen mit der *id* -1 aus *lane section* B in zwei OpenDRIVE-Fahrstreifen auf, die beispielsweise die Trennung eines Bürgersteigs in Geh- und Fahrradweg repräsentieren. Entsprechend wird die neue *lane section* C definiert. Da beim Übergang von *lane section* B zu *lane section* C keine Änderung der Verhaltensattribute auftritt, ist an dieser Stelle keine Definition eines neuen BSSD-Segments erforderlich.

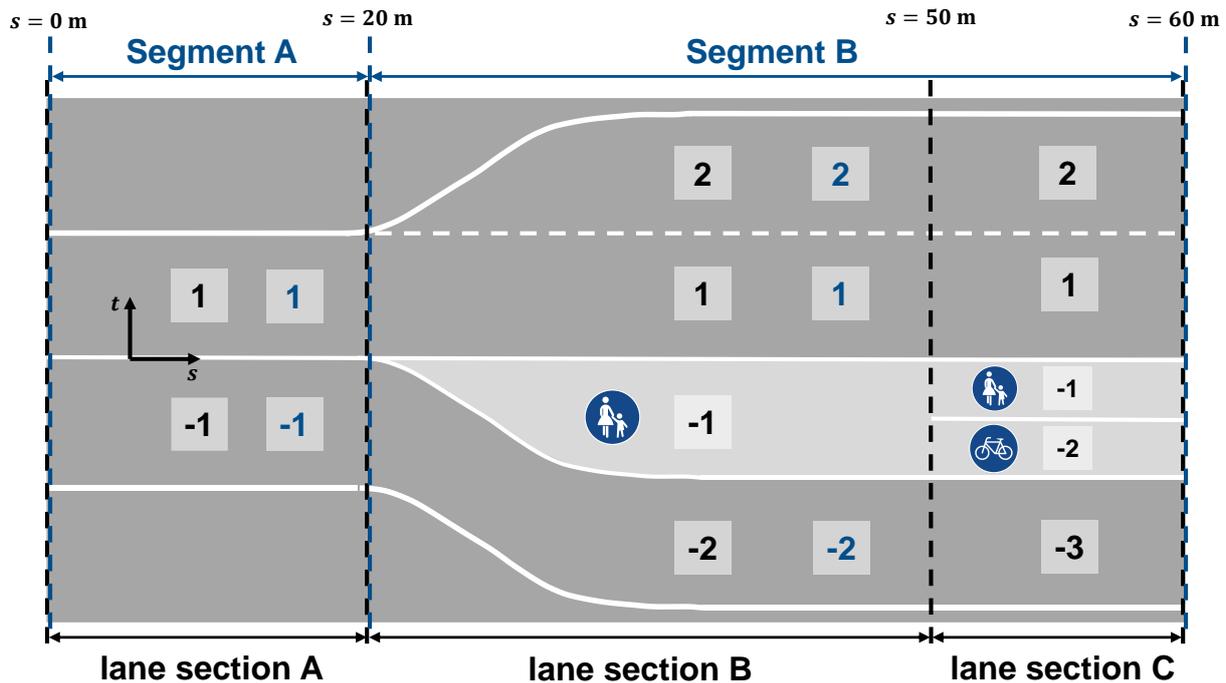


Abbildung 5-10: Beispiel Erstellung von BSSD-Fahrstreifen basierend auf OpenDRIVE-Fahrstreifen

Basierend auf den BSSD-Segmenten und den OpenDRIVE-Fahrstreifen werden anschließend die BSSD-Fahrstreifen erstellt. Der Definitionsbereich von Segment A überschneidet sich nur mit *lane section* A. Somit ist *lane section* A die erste zu Segment A überschneidende *lane section*. Resultierend aus der in diesem Abschnitt definierten Regel für die Erstellung von BSSD-Fahrstreifen werden in Segment A die zwei BSSD-Fahrstreifen mit der *id* -1 und 1 erstellt. Der Definitionsbereich von

Segment B überschneidet mit *lane section B* und *lane section C*. Da *lane section B* die erste mit Segment B überschneidende *lane section* ist, werden die BSSD-Fahrstreifen in Segment B basierend auf den OpenDRIVE-Fahrstreifen in dieser *lane section* definiert. Es resultieren drei BSSD-Fahrstreifen mit den *id*-Werten 2, 1 und -2 für das Segment B. Die in dem in Abbildung 5-10 dargestellten Beispiel enthaltenen OpenDRIVE-Fahrstreifen und *lane sections* sowie die daraus resultierenden BSSD-Fahrstreifen und -Segmente sind zusammenfassend in Tabelle 5-5 und Tabelle 5-6 gelistet.

Tabelle 5-5: *lane sections* und OpenDRIVE-Fahrstreifen basierend auf Abbildung 5-10

<b>&lt;laneSection&gt;</b>	<b>OpenDRIVE- &lt;lane&gt;</b>
<hr/>	
<i>s</i> = 0.0	
	<i>id</i> = 1
	<i>id</i> = -1
<hr/>	
<i>s</i> = 20.0	
	<i>id</i> = 2
	<i>id</i> = 1
	<i>id</i> = -1
	<i>id</i> = -2
<hr/>	
<i>s</i> = 50.0	
	<i>id</i> = 2
	<i>id</i> = 1
	<i>id</i> = -1
	<i>id</i> = -2
	<i>id</i> = -3

Tabelle 5-6: BSSD-Segmente und -Fahrstreifen basierend auf Abbildung 5-10

<b>&lt;segment&gt;</b>	<b>BSSD-&lt;lane&gt;</b>
<hr/>	
<i>sStart</i> = 0.0	
	<i>id</i> = 1
	<i>id</i> = -1
<hr/>	
<i>sStart</i> = 20.0	
	<i>id</i> = 2
	<i>id</i> = 1
	<i>id</i> = -2

Zur Umsetzung der in diesem Abschnitt definierten Regel für die Erstellung der BSSD-Fahrstreifen ist die Durchführung eines Algorithmus notwendig, der für jedes zuvor definierte BSSD-Segment alle mit diesem BSSD-Segment überschneidenden *lane sections* findet. Anschließend wird, falls mehrere mit dem BSSD-Segment überschneidende *lane sections* existieren, die erste dieser *lane sections* ausgewählt. Resultierend aus diesem Algorithmus existiert somit für jedes BSSD-Segment eine erste überschneidende *lane section*. Für jedes in dieser *lane section* enthaltene OpenDRIVE-<lane>-Element, das einen befahrbaren OpenDRIVE-Fahrstreifen repräsentiert, wird ein äquivalentes BSSD-<lane>-Element in dem BSSD-Segment erstellt.

### 5.3.1.6 Schritt 6: Verknüpfung der OpenDRIVE- & BSSD-Fahrstreifen

Im sechsten Schritt der BSSD-Integration wird eine explizite Verknüpfung zwischen den zuvor erstellen BSSD-Fahrstreifen und den OpenDRIVE-Fahrstreifen hergestellt. Für die Bestimmung des Verhaltensraums an einer definierten Position der OpenDRIVE-Karte ist erforderlich, dass jeder Stelle eines befahrbaren OpenDRIVE-Fahrstreifens eindeutig ein BSSD-Fahrstreifen zuzuordnen ist. Diese Zuordnung wird über eine innerhalb eines BSSD-Fahrstreifens definierte Hilfsstruktur realisiert.

Wie in Abschnitt 5.3.1.5 beschrieben, überschneidet immer mindestens eine *lane section* mit einem BSSD-Segment. Die OpenDRIVE- und BSSD-Fahrstreifen werden longitudinal durch eine *lane section* bzw. ein BSSD-Segment begrenzt. Die lateralen Begrenzungen der OpenDRIVE- und BSSD-Fahrstreifen sind aufgrund der Definition der BSSD-Fahrstreifen basierend auf den OpenDRIVE-Fahrstreifen identisch. Aufgrund dessen überschneidet immer mindestens ein OpenDRIVE-Fahrstreifen mit einem BSSD-Fahrstreifen.

Die angesprochene Hilfsstruktur zur eindeutigen Zuordnung von OpenDRIVE- und BSSD-Fahrstreifen wird anhand eines im Rahmen der BSSD-Integration neu definierten Elements `<linkedLane>` realisiert. Für jeden OpenDRIVE-Fahrstreifen, der mit einem BSSD-Fahrstreifen überschneidet, wird unterhalb des zugehörigen BSSD-`<lane>`-Elements ein Element `<linkedLane>` erstellt, das auf diesen OpenDRIVE-Fahrstreifen verweist. Dazu werden die dem `<linkedLane>`-Element zugehörigen Attribute `id` und `sLaneSection` definiert, welche die `id` des überschneidenden OpenDRIVE-Fahrstreifens sowie die `s`-Koordinate der *lane section*, die diesen OpenDRIVE-Fahrstreifen beinhaltet, angeben. Somit ist es möglich, jeder Position der OpenDRIVE-Karte, bei der ein befahrbarer OpenDRIVE-Fahrstreifen definiert ist, anhand der eindeutigen Kombination der `s`-Koordinate, der `id` des OpenDRIVE-Fahrstreifens und der `s`-Koordinate der *lane section* eindeutig einen BSSD-Fahrstreifen und folglich die an dieser Stelle gültigen Verhaltensattribute zuzuordnen. Zur Verdeutlichung der Verwendung des `<linkedLane>`-Elements sind in Tabelle 5-7 alle aus der in Abbildung 5-10 dargestellten Szenerie resultierenden `<linkedLane>`-Elemente sowie die dazugehörigen Attribute für jeden BSSD-Fahrstreifen gelistet.

Tabelle 5-7: Verknüpfung OpenDRIVE- und BSSD-Fahrstreifen basierend auf Abbildung 5-10

<code>&lt;segment&gt;</code>	BSSD- <code>&lt;lane&gt;</code>	<code>&lt;linkedLane&gt;</code> <code>sLaneSection</code>	<code>&lt;linkedLane&gt;</code> <code>id</code>
<code>sStart = 0.0</code>			
	<code>id = 1</code>		
		<code>0.0</code>	<code>1</code>
	<code>id = -1</code>		
		<code>0.0</code>	<code>-1</code>
<code>sStart = 20.0</code>			
	<code>id = 2</code>		
		<code>20.0</code>	<code>2</code>
		<code>50.0</code>	<code>2</code>
	<code>id = 1</code>		
		<code>20.0</code>	<code>1</code>
		<code>50.0</code>	<code>1</code>
	<code>id = -2</code>		
		<code>20.0</code>	<code>-2</code>
		<code>50.0</code>	<code>-3</code>

---

Für die Erstellung der in diesem Abschnitt beschriebenen Hilfsstruktur ist die Durchführung eines Algorithmus erforderlich, der für jeden BSSD-Fahrstreifen alle zu diesem BSSD-Fahrstreifen überschneidenden OpenDRIVE-Fahrstreifen findet. Zur Verbesserung der Übersichtlichkeit werden alle einem BSSD-Fahrstreifen zugeordneten `<linkedLane>`-Elemente einem Element `<assignLane-OpenDRIVE>` untergeordnet. Letzteres repräsentiert somit die Verknüpfung zwischen den BSSD- und OpenDRIVE-Fahrstreifen und wird genau einmal je BSSD-`<lane>`-Element definiert.

#### 5.3.1.7 Schritt 7: Erstellen der Verhaltensraum-Minimalstruktur

Im siebten Schritt der BSSD-Integration wird innerhalb von jedem BSSD-Fahrstreifen eine Verhaltensraum-Minimalstruktur eingefügt. Diese Minimalstruktur beinhaltet übergeordnet die beiden Elemente `<behaviorAlong>` und `<behaviorAgainst>`. Diesen Elementen untergeordnet werden jeweils alle bei der Definition eines atomaren Verhaltensraums entsprechend der Spezifikation der BSSD in jedem Fall verwendeten Elemente und Attribute eingefügt. Somit ist eine Grundlage geschaffen, um die Ausprägungen der Verhaltensattribute in den einzelnen BSSD-Fahrstreifen festzulegen. Die Verhaltensraum-Minimalstruktur ist im Anhang A.5, Abbildung A-2 gezeigt.

#### 5.3.1.8 Schritt 8: Befüllen von Verhaltensattributen

Im achten Schritt der BSSD-Integration werden für jeden BSSD-Fahrstreifen die innerhalb dieses Fahrstreifens geltenden Ausprägungen der Verhaltensattribute entsprechend der Definition der BSSD festgelegt. Dazu ist die Durchführung eines Algorithmus erforderlich, der basierend auf den in der OpenDRIVE-Karte hinterlegten Informationen die Ausprägungen der BSSD-Verhaltensattribute automatisiert bestimmt. Die Ausprägungen der Verhaltensattribute werden anschließend in die zuvor erstellte Minimalstruktur des Verhaltensraums an der passenden Stelle eingefügt.

Es sei angemerkt, dass das vollständige Bestimmen der Ausprägungen der BSSD-Verhaltensattribute kein Teil der Aufgabenstellung dieser Arbeit ist. Da das in diesem Abschnitt beschriebene Konzept das Ziel einer gesamtheitlichen Integration der BSSD in eine OpenDRIVE-Karte verfolgt, ist dieser Konzept-Schritt dennoch im Konzept enthalten. Der für die Durchführung dieses Konzept-Schritts notwendige Prozess des automatisierten Ableitens von Verhaltensattributen ist näher bei Glatzki et al.<sup>72</sup> beschrieben.

#### 5.3.1.9 Schritt 9: Abspeichern der modifizierten OpenDRIVE-Datei

Im letzten Schritt der BSSD-Integration wird die durch die vorigen Konzept-Schritte modifizierte OpenDRIVE-Datei abgespeichert. Somit existiert bei vollständiger und korrekter Durchführung aller zuvor beschriebenen Konzept-Schritte eine OpenDRIVE-Datei, die um eine Struktur ergänzt ist, welche alle BSSD-Informationen zu der in der OpenDRIVE-Karte dargestellten Szenerie enthält.

Alle in diesem Abschnitt abgeleiteten Konzept-Schritte sowie die für die Umsetzung dieser Schritte benötigten Algorithmen sind zusammenfassend in Abbildung 5-11 dargestellt. Es sei erneut darauf

---

<sup>72</sup> Glatzki, F.; Winner, H.: Inferenz von Verhaltensattributen der verhaltenssemantischen Szeneriebeschreibung (2022).

hingewiesen, dass das in diesem Abschnitt abgeleitete Konzept eine ideale Vorgehensweise darstellt, deren Umsetzbarkeit im Rahmen dieser Arbeit zunächst zu bewerten ist (siehe Abschnitt 5.3.2).

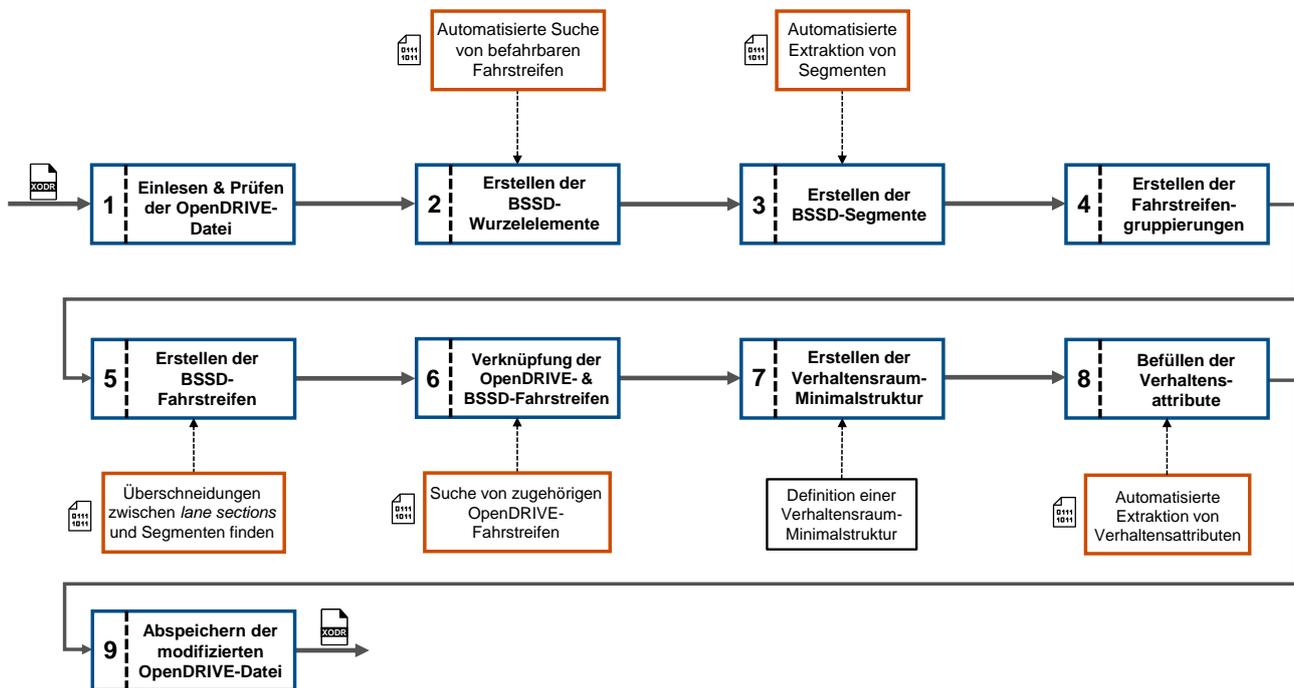


Abbildung 5-11: Schritte und Algorithmen für eine ideale Realisierung von Konzept 2

## 5.3.2 Umsetzungsbewertung

Wie zuvor beschrieben, ist es möglich, dass einzelne Konzept-Schritte nicht oder nur zu Teilen realisierbar sind. Aufgrund dessen werden in diesem Abschnitt die einzelnen zur Umsetzung des ausdetaillierten Konzepts notwendigen Schritte (siehe Abbildung 5-11) bezüglich einer Umsetzung bewertet. Dazu wird für jeden Konzept-Schritt eine Einschätzung gegeben, ob die Umsetzung des jeweiligen Schritts realistisch ist. Diese Einschätzungen basieren auf dem Wissenstand vor der Implementierung bezüglich möglicherweise bei der Umsetzung auftretenden Schwierigkeiten und Einschränkungen, die aus dem Informationsgehalt einer OpenDRIVE-Karte resultieren.

### 5.3.2.1 Bewertung Schritt 1 – Einlesen & Prüfen der OpenDRIVE-Datei

Es existieren offizielle XML-Schema-Dateien für die Versionsveröffentlichungen von OpenDRIVE. Daher ist von einem geringen Aufwand für die Realisierung der Überprüfung der Validität einer vorhandenen OpenDRIVE-Datei auszugehen. Aufgrund dessen ist die vollständige Umsetzung des ersten Konzept-Schritts „Einlesen & Prüfen der OpenDRIVE-Datei“ realistisch.

### 5.3.2.2 Bewertung Schritt 2 – Erstellen der BSSD-Wurzelemente

Bei Betrachtung der Umsetzung des zweiten Konzept-Schritts erscheint es möglich, dass ein Algorithmus zur automatisierten Suche von befahrbaren Fahrstreifen, einzelne OpenDRIVE-Fahrstreifen falsch als befahrbar bzw. nicht-befahrbar einteilt. Die Ursache dafür liegt in dem begrenzten Informationsgehalt der OpenDRIVE-Karte bezüglich der Fahrbahnzugehörigkeit der einzelnen

---

OpenDRIVE-Fahrstreifen. Ein OpenDRIVE-Fahrstreifen besitzt das verpflichtende Attribut `type`, das die Art des OpenDRIVE-Fahrstreifens definiert (siehe Abschnitt 3.3.1.3). Basierend auf diesem Attribut ist eine automatisierte Klassifizierung umsetzbar, die jedoch möglicherweise Fehler aufweist. Beispielsweise ist es möglich, dass das Attribut `type="biking"` einen Fahrradschutzstreifen repräsentiert, der einen Teil der Fahrbahn darstellt, und somit eine Modellierung mit der BSSD erfordert. Andererseits ist es jedoch möglich, dass ein baulich von der Fahrbahn getrennter Fahrradweg, der keinen Teil der Fahrbahn darstellt, ebenfalls mit dem Attribut `type="biking"` modelliert wird (siehe Abbildung 6-5). In diesem Fall wäre somit keine Modellierung mit der BSSD erforderlich.

Um das möglicherweise auftretende Problem der falschen Einordnung von OpenDRIVE-Fahrstreifen als befahrbare bzw. nicht-befahrbare OpenDRIVE-Fahrstreifen zu lösen, wird die Umsetzung eines zweiten Algorithmus vorgesehen. Dieser zweite Algorithmus ermöglicht das manuelle Bearbeiten der aus dem vorhergehenden Algorithmus resultierenden Einteilung der OpenDRIVE-Fahrstreifen in befahrbare und nicht-befahrbare OpenDRIVE-Fahrstreifen anhand einer Nutzereingabe. Somit ist es möglich, falsche Ergebnisse der automatisierten Suche von befahrbaren Fahrstreifen zu korrigieren. Resultierend aus der Ergänzung des Konzept-Schritts „Erstellen der BSSD-Wurzelemente“ um den beschriebenen Algorithmus für das manuelle Bearbeiten der befahrbaren Fahrstreifen, ist eine vollständige Umsetzung dieses Konzept-Schritts realistisch.

### 5.3.2.3 Bewertung Schritt 3 – Erstellen der BSSD-Segmente

Bei Betrachtung der Umsetzung des dritten Konzept-Schritts erscheint es wahrscheinlich, dass in vielen Anwendungsfällen nicht alle BSSD-Segmente basierend auf Grundregeln, welche die in der OpenDRIVE-Karte bereits enthaltenen BSSD-relevanten Informationen nutzen (siehe Abschnitt 5.3.1.3), automatisiert gefunden werden. Die Ursache dafür ist der begrenzte BSSD-relevante Informationsgehalt der OpenDRIVE-Karte. Die Einteilung einer Straße in BSSD-Segmente ist abhängig vom Zustand der Verhaltensattribute im longitudinalen Verlauf der Straße. Da der Zustand der Verhaltensattribute in den Straßen einer gegebenen OpenDRIVE-Karte im Allgemeinen bekannt ist<sup>73</sup>, ist es meist nicht möglich, die zu einer OpenDRIVE-Karte zugehörigen BSSD-Segmente vollständig automatisiert aus der Karte zu extrahieren.

Um dennoch eine vollständige aus der Szenerie der OpenDRIVE-Karte resultierende Segmentierung zu erreichen, wird die Umsetzung eines zweiten Algorithmus vorgesehen, der das manuelle Bearbeiten der automatisiert gefundenen BSSD-Segmente anhand einer Nutzereingabe ermöglicht. Somit ist anhand der Kombination des Algorithmus zur automatisierten Extraktion der BSSD-Segmente aus der OpenDRIVE-Karte und des Algorithmus zum manuellen Bearbeiten von BSSD-Segmenten möglich, eine vollständige und korrekte Segmentierung einer OpenDRIVE-Karte vorzunehmen. Resultierend aus der Ergänzung des Konzept-Schritts „Erstellen der BSSD-Segmente“ um den beschriebenen Algorithmus für das manuelle Bearbeiten von BSSD-Segmenten ist eine vollständige Umsetzung dieses Konzept-Schritts realistisch.

---

<sup>73</sup> Dieser Zustand würde den idealen Zielzustand des in dieser Arbeit definierten Konzepts darstellen

---

#### 5.3.2.4 Bewertung Schritt 4 – Erstellen der Fahrstreifen-Gruppierungen

Der vierte Konzept-Schritt „Erstellen der Fahrstreifen-Gruppierungen“ dient ausschließlich der Verbesserung der menschlichen Lesbarkeit der OpenDRIVE-Datei. Somit ist die vollständige Umsetzung dieses Konzept-Schritts realistisch.

#### 5.3.2.5 Bewertung Schritt 5 – Erstellen der BSSD-Fahrstreifen

Eine Schwierigkeit bei der Umsetzung des für den fünften Konzept-Schritt benötigten Algorithmus zum Finden der Überschneidungen zwischen *lane sections* und BSSD-Segmenten ist die Beachtung des Spezialfalls von einseitig definierten *lane sections* (Attribut `singleSide="true"`). Dieser Spezialfall ermöglicht, dass bei einer *s*-Koordinate zwei *lane sections* definiert sind und somit bei bestimmten Bereichen der *s*-Koordinate zwei *lane sections* gleichzeitig mit einem BSSD-Segment überschneiden. Da dieser Spezialfall nach dem Wissenstand vor der Implementierung die Umsetzung des Konzepts nicht verhindert, ist die vollständige Umsetzung des fünften Konzept-Schritts „Erstellen der BSSD-Fahrstreifen“ realistisch.

#### 5.3.2.6 Bewertung Schritt 6 – Verknüpfen der BSSD- & OpenDRIVE-Fahrstreifen

Die Schwierigkeit bei der Umsetzung des sechsten Konzept-Schritts ergibt sich analog zur Umsetzungsbewertung bezüglich des vorigen Konzept-Schritts durch den Spezialfall von einseitig definierten *lane sections*. Bei der Suche eines zu einem BSSD-Fahrstreifen überschneidenden OpenDRIVE-Fahrstreifens ist somit erforderlich, diesen Spezialfall zu berücksichtigen. Analog zur Umsetzungsbewertung des vorigen Konzept-Schritts ist die vollständige Umsetzung des sechsten Konzept-Schritts „Verknüpfen der BSSD- & OpenDRIVE-Fahrstreifen“ dennoch realistisch.

#### 5.3.2.7 Bewertung Schritt 7 – Erstellen der Verhaltensraum-Minimalstruktur

Im siebten Konzept-Schritt wird in jeden BSSD-Fahrstreifen eine vordefinierte XML-Struktur eingefügt. Somit ist die vollständige Umsetzung dieses Konzept-Schritts realistisch.

#### 5.3.2.8 Bewertung Schritt 8 – Befüllen der Verhaltensattribute

Im achten Konzept-Schritt werden in jedem BSSD-Fahrstreifen die Ausprägungen der Verhaltensattribute automatisiert aus der OpenDRIVE-Karte extrahiert. Wie bereits erwähnt, ist das Ableiten der Ausprägungen der BSSD-Verhaltensattribute kein Teil der Aufgabenstellung dieser Arbeit.

Wie Glatzki et al.<sup>74</sup> zeigen, ist es grundsätzlich möglich, die Ausprägungen der Verhaltensattribute automatisiert zu bestimmen. Dies wird im Rahmen dieser Arbeit mit dem Verhaltensattribut *Speed Limit*, das einen einfachen Informationsgehalt besitzt, exemplarisch demonstriert. In Abhängigkeit des Informationsgehalts der verwendeten OpenDRIVE-Karte ist davon auszugehen, dass die Ausprägungen des Verhaltensattributs *Speed Limit* automatisiert zu Teilen oder vollständig bestimmbar sind. Daraus resultierend wird der achte Konzept-Schritt im Folgenden als „Befüllen des

---

<sup>74</sup> Glatzki, F.; Winner, H.: Inferenz von Verhaltensattributen der verhaltenssemantischen Szeneriebeschreibung (2022).

Verhaltensattributs *Speed Limit*“ sowie der für die Durchführung dieses Schritts notwendige Algorithmus als „Automatisierte Extraktion des Verhaltensattributs *Speed Limit*“ bezeichnet. Somit ist eine vollständige Umsetzung dieses Konzept-Schritts im Rahmen dieser Arbeit möglich.

### 5.3.2.9 Bewertung Schritt 9 – Abspeichern der modifizierten OpenDRIVE-Datei

Der letzte Konzept-Schritt „Abspeichern der modifizierten OpenDRIVE-Datei“ ist bei Durchführung der vorigen Konzept-Schritte in einer Programmiersprache vollständig realisierbar.

### 5.3.2.10 Zusammenfassung

Zusammenfassend erscheint die vollständige Umsetzung des in Abschnitt 5.3.1 ausdetaillierten Konzepts zur Integration der BSSD in OpenDRIVE mit den in diesem Abschnitt beschriebenen Modifikationen realistisch. Bei den beiden Konzept-Schritten „Erstellen der BSSD-Wurzelemente“ und „Erstellen der BSSD-Segmente“ ist es nicht möglich, die Richtigkeit bzw. Vollständigkeit der automatisiert gefundenen Ergebnisse in jedem Anwendungsfall zu garantieren. Die Ursache dafür ist der begrenzte Informationsgehalt der OpenDRIVE-Karte. Aufgrund dessen ist für die vollständige Umsetzung dieser beiden Konzept-Schritte jeweils eine Erweiterung um einen Algorithmus zum manuellen Bearbeiten der automatisiert erzeugten Ergebnisse erforderlich. Die Realisierung des Konzept-Schritts „Befüllen der Verhaltensattribute“ ist nicht Gegenstand dieser Arbeit. Eine exemplarische Demonstration der Umsetzbarkeit dieses Konzept-Schritts wird dennoch angestrebt. Daher wird die Bezeichnung für diesen Konzept-Schritt sowie den dafür benötigten Algorithmus angepasst.

Das um die Änderungen, die aus der Umsetzungsbewertung resultieren, erweiterte Konzept ist zusammenfassend in Abbildung 5-12 dargestellt. Im Unterschied zu dem in Abbildung 5-11 dargestellten idealen Konzept, sind alle Schritte realistisch im Rahmen dieser Arbeit umsetzbar. Das dargestellte Konzept stellt die Grundlage für die Implementierung dar.

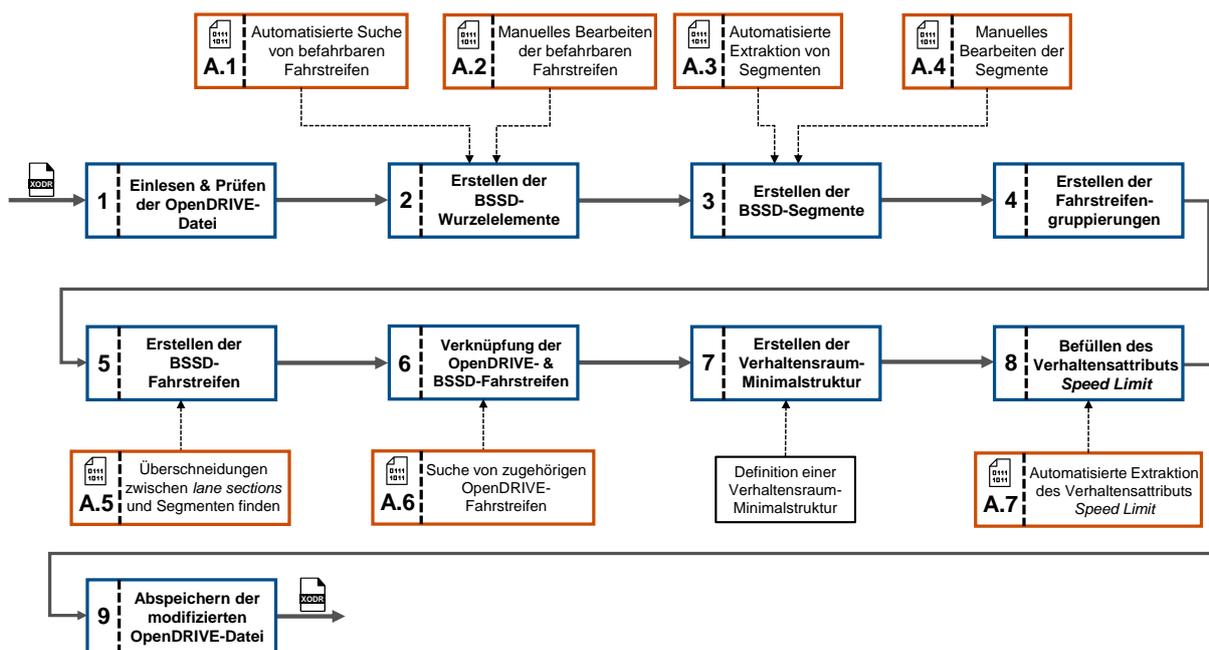


Abbildung 5-12: Schritte und Algorithmen für eine umsetzbare Realisierung von Konzept 2

---

## 6 Implementierung

---

In diesem Kapitel wird die Implementierung des im vorigen Kapitel entwickelten Konzepts zur ganzheitlichen Integration der BSSD in eine OpenDRIVE-Karte beschrieben. Dazu wird in Abschnitt 6.1 eine Programmiersprache für die Implementierung begründet ausgewählt. Die der Implementierung zugrundeliegende Dateistruktur wird in Abschnitt 6.2 beschrieben. Die Implementierung der einzelnen Konzept-Schritte wird in Abschnitt 6.3 dargestellt. In Abschnitt 6.4 wird das Testen der implementierten Komponenten beschrieben.

### 6.1 Auswahl der Programmiersprache

Das Konzept wird als ein in einer Konsole ausführbares Software-Tool in der Programmiersprache *Python* umgesetzt. *Python* steht öffentlich sowie kostenlos zur Verfügung (Open Source) und ist in den gängigsten Betriebssystemen (Windows, MacOS und Linux) ausführbar. Somit ist eine hohe Kompatibilität des Software-Tools gewährleistet. Darüber hinaus existieren für *Python* viele Bibliotheken, die das Verarbeiten von großen Mengen an Daten und Dateien, wie es in dieser Arbeit notwendig ist, auf einfachem Weg ermöglichen. Im Speziellen für die Verarbeitung von OpenDRIVE-Dateien existiert das *Python*-Modul `opendrive_parser`, das eine Teilfunktionalität der öffentlich zugänglichen Toolbox *CommonRoad Scenario Designer*<sup>75</sup> von Maierhofer et al.<sup>76</sup> ist. Dieses Modul nimmt eine Objekterstellung basierend auf einer OpenDRIVE-Datei vor und stellt somit eine mögliche Grundlage für die Implementierung des Konzepts dar. Aus den genannten Gründen wird die Programmiersprache *Python* für die Implementierung verwendet.

### 6.2 Dateistruktur

Die grundlegende Struktur des Software-Tools wird anhand von Abbildung 6-1 erklärt, welche die Ordnerstruktur des Software-Tools zeigt. Der Haupt-Bestandteil des Software-Tools wird durch den Ordner `integrate_BSSD_into_OpenDRIVE` repräsentiert. Dieser Ordner beinhaltet vier Unterordner, die jeweils unterschiedliche Funktionen beinhalten. Der Unterordner `concept_steps` beinhaltet für die Durchführung eines jeden Konzept-Schritts eine separate *Python*-Funktion. Bei der Durchführung der Konzept-Schritte werden die dazu benötigten Algorithmen aufgerufen. Diese sind im Unterordner `algorithms` enthalten und werden ebenfalls jeweils durch eine separate *Python*-Funktion repräsentiert. Der Unterordner `opendrive_parser` beinhaltet das gleichnamige *Python*-Modul, das der Erzeugung von Objekten basierend auf den Elementen der eingelesenen OpenDRIVE-Datei dient. Im Unterordner `utility` befinden sich Hilfsfunktionen, die an unterschiedlichen Stellen bei der Ausführung der Konzept-Schritte und Algorithmen benötigt werden. Die eigentliche Ausführung des Software-Tools erfolgt mit der Funktion `main.py`, die alle Konzept-Schritte aus dem Unterordner `concept_steps` inkrementell aufruft.

---

<sup>75</sup> <https://gitlab.lrz.de/tum-cps/commonroad-scenario-designer>

<sup>76</sup> Maierhofer, S. et al.: CommonRoad Scenario Designer: An Open-Source Toolbox for Map Conversion (2021).

Der Ordner `data` beinhaltet alle Daten, die zur Ausführung des Software-Tools notwendig sind. Im Unterordner `xodr_files` werden OpenDRIVE-Dateien abgelegt, die entsprechend des Konzepts verarbeitet werden. Diese repräsentieren somit den Eingang des Software-Tools. Nach Verarbeitung einer OpenDRIVE-Datei durch das Software-Tool wird diese im Unterordner `BSSD` abgelegt. Die in diesem Ordner enthaltenen OpenDRIVE-Dateien repräsentieren somit den Ausgang des Software-Tools. Der Ordner `xsd_files` beinhaltet die offiziellen XML-Schema-Dateien von OpenDRIVE. Diese werden zur Überprüfung der Validität der importierten OpenDRIVE-Dateien verwendet (siehe Abschnitt 6.3.1). Die XML-Datei `minimal_structure_BSSD_behavior_space.xml` definiert die bei Ausführung von Konzept-Schritt 7 „Erstellen der Verhaltensraum-Minimalstruktur“ benötigte Minimalstruktur eines Verhaltensraums der BSSD.

Der Ordner `test` beinhaltet Funktionen, die jeweils einzelne Testfälle repräsentieren. Diese dienen dem Testen der korrekten Funktionalität der im Ordner `integrate_BSSD_into_OpenDRIVE` enthaltenen Funktionen (siehe Abschnitt 6.4). Die für die Ausführung der Testfälle benötigten Daten sind im Unterordner `test_data` enthalten, der analog zur Struktur des Ordners `data` die Unterordner `xodr_files` und `xsd_files` beinhaltet. Der Unterordner `utility` beinhaltet Hilfsfunktionen, die beim Ausführen der Testfälle benötigt werden.

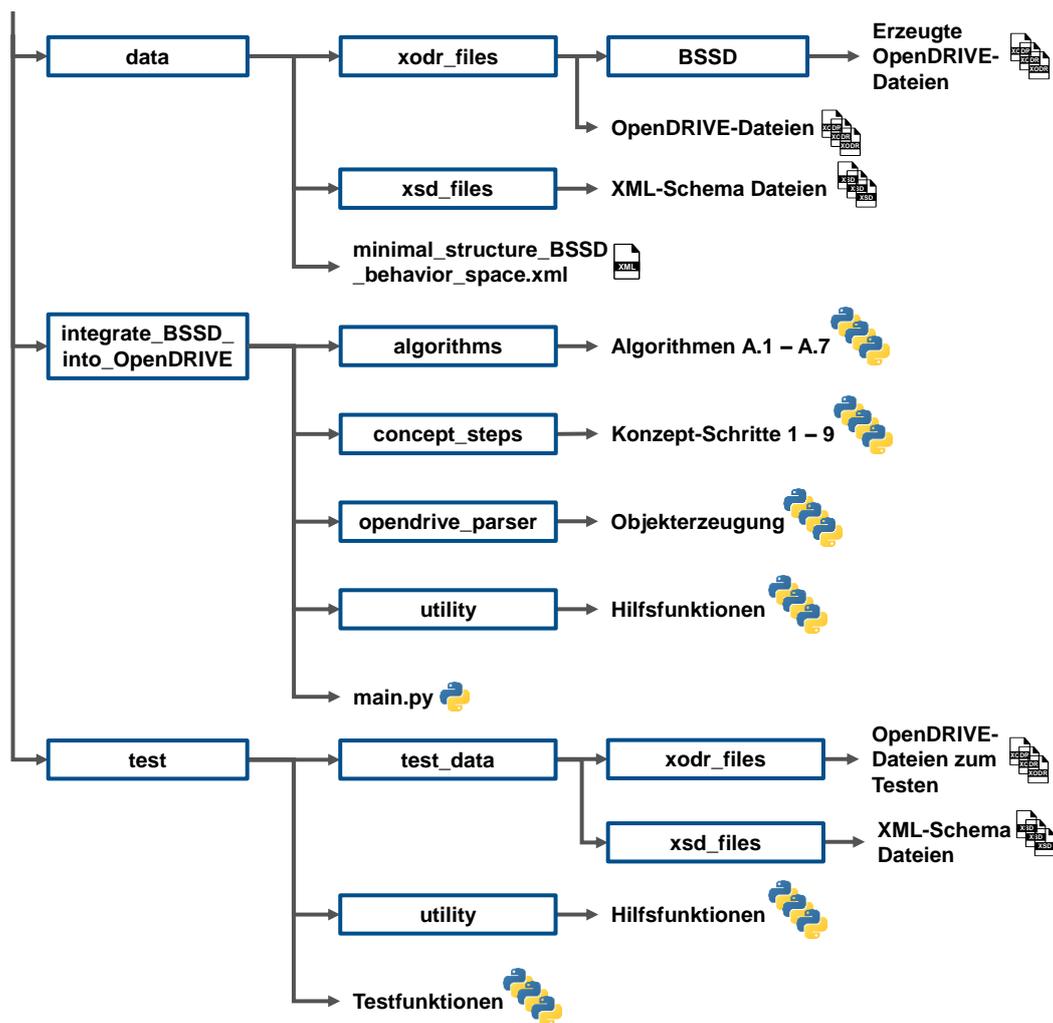


Abbildung 6-1: Struktur des Software-Tools

## 6.3 Umsetzung der Konzept-Schritte

In diesem Abschnitt wird die Umsetzung der einzelnen Konzept-Schritte beschrieben. Wie zuvor erwähnt, erfolgt die Ausführung des Software-Tools durch das Skript `main.py`, das alle Konzept-Schritte inkrementell aufruft. Der daraus resultierende Ablauf des Software-Tools ist in Abbildung 6-3 dargestellt, die ein Gesamtschaubild der Implementierung repräsentiert. Analog zum Schaubild des ausdetaillierten Konzepts (siehe Abbildung 5-12) sind die Konzept-Schritte und Algorithmen als blaue bzw. orangene Rechtecke dargestellt. Einzelne Dateien der Dateiformate XODR, XSD und XML werden durch Datei-Icons dargestellt. Die zentralen für die Implementierung des Software-Tools verwendeten externen *Python*-Module sind in grünen Rechtecken dargestellt. Die beschrifteten Pfeile repräsentieren einzelne Datenstrukturen, die entweder aus einer Datei oder dem Ausgang eines Konzept-Schritts bzw. Algorithmus resultieren. Die strichlinierten Pfeile repräsentieren Funktionalitäten, die durch ein externes *Python*-Modul bereitgestellt werden.

### 6.3.1 Schritt 1: Einlesen und Prüfen der OpenDRIVE-Datei

Der erste Konzept-Schritt „Einlesen & Prüfen der OpenDRIVE-Datei“ dient dem Einlesen der zu verarbeitenden OpenDRIVE-Datei sowie der Überprüfung der Validität dieser. Das Funktionsschaubild des ersten Konzept-Schritts ist in Abbildung 6-2 dargestellt, die den zu diesem Schritt zugehörigen Teil des Gesamtschaubilds der Implementierung beinhaltet.

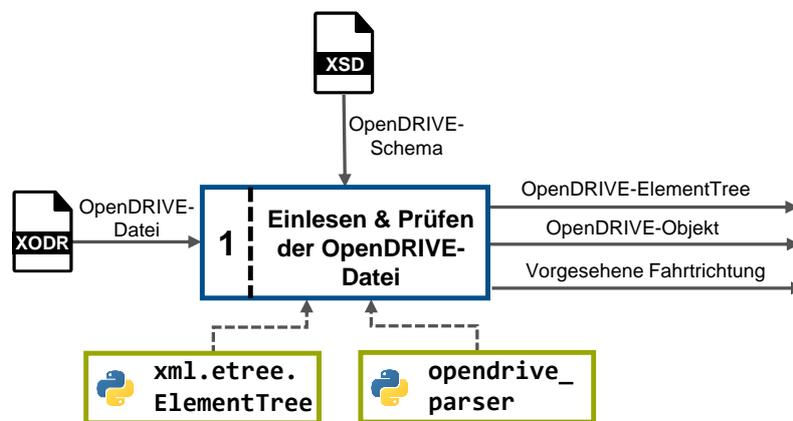


Abbildung 6-2: Funktionsschaubild Schritt 1 „Einlesen & Prüfen der OpenDRIVE-Datei“

Der Eingang des ersten Konzept-Schritts ist eine externe OpenDRIVE-Datei (Dateiformat XODR), deren Verarbeitung entsprechend des Konzepts erwünscht ist. Da eine OpenDRIVE-Datei gemäß des Dateiformats XML strukturiert ist, wird das standardmäßig in *Python* enthaltene Modul `xml.etree.ElementTree`<sup>77</sup> verwendet, um die OpenDRIVE-Datei einzulesen. Dieses Modul ermöglicht das im Rahmen dieser Implementierung erwünschte Bearbeiten und Abspeichern der OpenDRIVE-Datei. Die aus dem Einlesen der OpenDRIVE-Datei mittels des Moduls `xml.etree.ElementTree` resultierende Datenstruktur wird im Folgenden als *OpenDRIVE-ElementTree* (siehe Abbildung 6-2) bezeichnet.

<sup>77</sup> Python Software Foundation: `xml.etree.ElementTree` — The ElementTree XML API (2022).

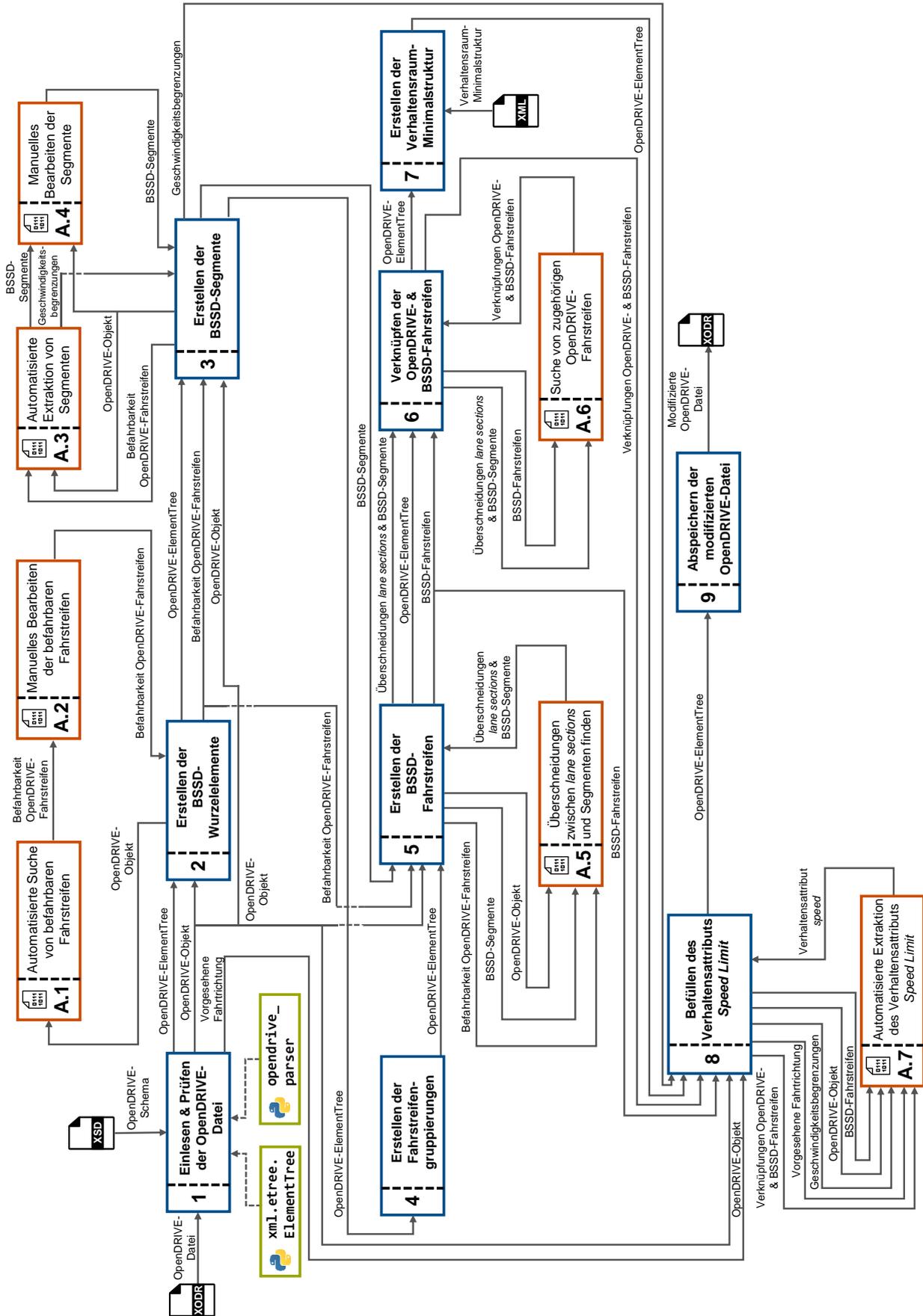


Abbildung 6-3: Gesamtschaubild der Implementierung

---

Nach dem Einlesen der OpenDRIVE-Datei wird die Validität der Datei überprüft. Dazu wird die Funktionalität eines externen *Python*-Moduls verwendet, das die Überprüfung der Validität einer XML-Datei entsprechend eines gegebenen XML-Schemas ermöglicht. In diesem Fall entspricht die XML-Datei dem OpenDRIVE-*ElementTree* sowie das XML-Schema einer offiziellen XML-Schema-Datei von ASAM. Die Implementierung des Software-Tools wird mit dem Ziel durchgeführt, die Verarbeitung von OpenDRIVE-Dateien der Versionen 1.4, 1.5, 1.6 und 1.7 zu ermöglichen. Dementsprechend wird die Validität der importierten OpenDRIVE-Datei mit den XML-Schema-Dateien der genannten Versionen von OpenDRIVE überprüft. Die importierte OpenDRIVE-Datei wird als valide eingestuft, sofern diese mindestens entsprechend eines XML-Schemas der genannten Versionen definiert ist. Falls die importierte OpenDRIVE-Datei invalide ist, ist es nicht möglich, die korrekte und vollständige Durchführung der Implementierung entsprechend des Konzepts zu garantieren.

Es hat sich gezeigt, dass einzelne OpenDRIVE-Dateien geringfügige Abweichungen von den Vorgaben der XML-Schema-Dateien, wie beispielweise ein fehlendes Attribut, aufweisen, die jedoch keine Auswirkungen auf die korrekte und vollständige Durchführung der Implementierung haben. Aufgrund dessen wird die Möglichkeit geboten, dass eine invalide OpenDRIVE-Datei dennoch mit dem Software-Tool weiterverarbeitet wird. Dazu wird eine Nutzereingabe realisiert, die im Falle einer invaliden OpenDRIVE-Datei abfragt, ob die Weiterverarbeitung der eingelesenen OpenDRIVE-Datei dennoch erwünscht ist.

Für den Fall, dass die eingelesene OpenDRIVE-Datei akzeptiert wurde (valide bzw. invalide, aber durch Nutzereingabe bestätigt), wird der OpenDRIVE-*ElementTree* entsprechend einer weiterentwickelten Version des externen *Python*-Moduls `opendrive_parser` verarbeitet. Dieses Modul dient der Erstellung von Objekten mit Attributen entsprechend der importierten OpenDRIVE-Datei. Für jedes XML-Element des OpenDRIVE-*ElementTree* wird ein Objekt erstellt, das Attribute entsprechend den zu diesem XML-Element zugehörigen Attributen besitzt. Daraus resultierend wird anhand des OpenDRIVE-*ElementTree* ein übergeordnetes Objekt erzeugt, das im Folgenden als *OpenDRIVE-Objekt* bezeichnet wird. Dieses OpenDRIVE-Objekt repräsentiert das Element `<OpenDRIVE>` (siehe Abbildung 3-1) der importierten OpenDRIVE-Datei. Es enthält Verweise auf die Objekte, welche jeweils die einzelnen Straßen und Kreuzungen der OpenDRIVE-Karte repräsentieren. Die den Straßen bzw. Kreuzungen zugehörigen Objekte besitzen wiederum Verweise auf die Objekte, welche die dem `<road>`- bzw. `<junction>`-Element untergeordneten Elemente repräsentieren. Entsprechend dieser Logik ist die gesamte hierarchische Struktur der importierten OpenDRIVE-Datei im OpenDRIVE-Objekt abgebildet. Letzteres ermöglicht somit die effiziente Verarbeitung der in der OpenDRIVE-Karte enthaltenen Informationen in den nachfolgenden Konzept-Schritten.

Für diese Arbeit wird das Modul `opendrive_parser` in der Version 0.3 des CommonRoad Scenario Designer<sup>78</sup> verwendet. Dieses Modul wurde basierend auf OpenDRIVE in der Version 1.4 implementiert und unterstützt darüber hinaus nur einen Teil aller in der OpenDRIVE-Datei möglicherweise vorhandenen Elemente und Attribute. Aufgrund dessen wird das Modul manuell um weitere

---

<sup>78</sup> [https://gitlab.lrz.de/tum-cps/commonroad-scenario-designer/-/tree/release\\_0.3](https://gitlab.lrz.de/tum-cps/commonroad-scenario-designer/-/tree/release_0.3)

Komponenten erweitert, um eine Verarbeitung von allen in dieser Arbeit benötigten Elementen und Attributen von OpenDRIVE zu ermöglichen, sowie die Kompatibilität bis zu der Version 1.7 von OpenDRIVE sicherzustellen.

Im letzten Teil des ersten Konzept-Schritts wird versucht, die vorgesehene Fahrtrichtung (Rechts- oder Linksverkehr) der in der importierten OpenDRIVE-Karte dargestellten Szenerie zu extrahieren. Diese Information wird bei der Ausführung des achten Konzept-Schritts „Befüllen des Verhaltensattributs *Speed Limit*“ (siehe Abschnitt 6.3.8) benötigt. Die vorgesehene Fahrtrichtung wird bei einer OpenDRIVE-Karte der Version 1.5 oder neuer im optionalen Attribut *rule* des <road>-Elements angegeben, das entsprechend Rechts- oder Linksverkehr die Werte "RHT" oder "LHT" annimmt. Da die Verwendung dieses Attributs optional ist und das Attribut bei einer OpenDRIVE-Karte der Version 1.4 nicht existiert, wird die vorgesehene Fahrtrichtung standardmäßig auf Rechtsverkehr gesetzt, falls aus der Karte keine Information bezüglich der vorgesehenen Fahrtrichtung extrahierbar ist. Für den Fall, dass Linksverkehr gilt und in der Karte keine Information bezüglich der vorgesehenen Fahrtrichtung hinterlegt ist, ist somit erforderlich, die vorgesehene Fahrtrichtung manuell auf Linksverkehr zu setzen.

### 6.3.2 Schritt 2: Erstellen der BSSD-Wurzelemente

Im zweiten Konzept-Schritt „Erstellen der BSSD-Wurzelemente“ wird in jeder Straße, die mindestens einen befahrbaren OpenDRIVE-Fahrsstreifen besitzt, ein Element <userData> erstellt (siehe Abschnitt 5.3.1.2). Das Funktionsschaubild des zweiten Konzept-Schritts ist in Abbildung 6-4 dargestellt, die den zu diesem Schritt zugehörigen Teil des Gesamtschaubilds der Implementierung beinhaltet.

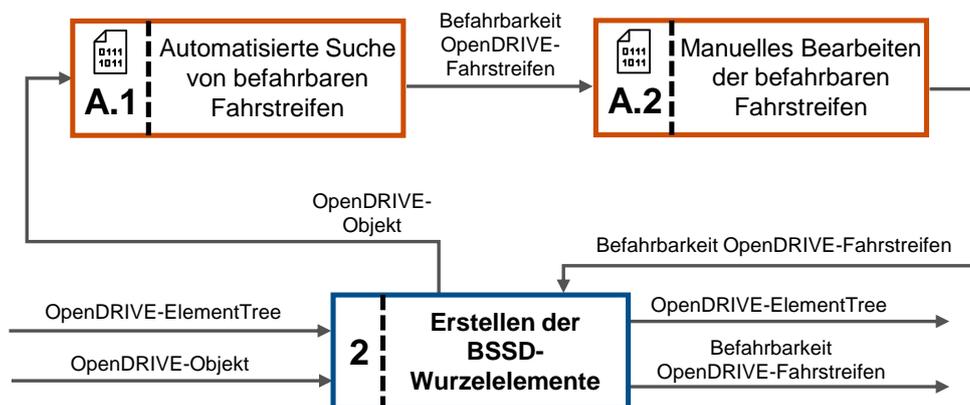


Abbildung 6-4: Funktionsschaubild Schritt 2 „Erstellen der BSSD-Segmente“

Die Implementierung des zweiten Konzept-Schritts besteht aus drei aufeinanderfolgenden Teilen, die im Folgenden näher erläutert werden.

#### 6.3.2.1 Algorithmus A.1: Automatisierte Suche von befahrbaren Fahrsstreifen

Der erste Teil des zweiten Konzept-Schritts ist die Durchführung des Algorithmus A.1 „Automatisierte Suche von befahrbaren Fahrsstreifen“. Bei diesem Algorithmus wird durch jeden OpenDRIVE-

Fahrstreifen der OpenDRIVE-Karte iteriert, um diesen jeweils als *befahrbar* oder *nicht-befahrbar* entsprechend der Zugehörigkeit zur Fahrbahn zu klassifizieren (siehe Abschnitt 5.3.1.2). Die Klassifizierung erfolgt anhand des verpflichtenden Attributs *type*, das die Art des Fahrstreifens angibt. Dazu wird für jede mögliche Ausprägung des *type*-Attributs eines OpenDRIVE-Fahrstreifens definiert, ob diese einen befahrbaren oder nicht-befahrbaren OpenDRIVE-Fahrstreifen repräsentiert. Alle entsprechend der Spezifikationen von OpenDRIVE möglichen Ausprägungen des *type*-Attributs (Versionen 1.4–1.7), deren Beschreibung sowie deren Auswirkung auf die Klassifizierung der Befahrbarkeit eines OpenDRIVE-Fahrstreifens sind im Anhang A.3, Tabelle A-5 gelistet.

OpenDRIVE-Fahrstreifen, die einen Fahrradweg repräsentieren (Attribut *type*="biking"), stellen einen Sonderfall dar und werden in Abweichungen zu OpenDRIVE-Fahrstreifen mit anderen *type*-Attributen zusätzlich anhand der benachbarten OpenDRIVE-Fahrstreifen als befahrbar bzw. nicht-befahrbar klassifiziert. Ein Fahrradweg wird grundsätzlich als nicht-befahrbar klassifiziert. Ist jedoch mindestens einer der beiden benachbarten OpenDRIVE-Fahrstreifen, sofern dieser existiert, als befahrbar klassifiziert, wird der Fahrradweg ebenfalls als befahrbar klassifiziert. Diese Unterscheidung verfolgt das Ziel, auf der Fahrbahn befindliche Fahrradwege als befahrbar zu klassifizieren und Fahrradwege, die getrennt von der Fahrbahn sind, als nicht-befahrbar zu klassifizieren. Zur Verdeutlichung dieses Sachverhalts sind in Abbildung 6-5 zwei Szenarien gezeigt. Die linke Szenerie beinhaltet einen Fahrradweg, der sich direkt auf der Fahrbahn befindet und somit einen befahrbaren OpenDRIVE-Fahrstreifen darstellt. In der rechten Szenerie ist der Fahrradweg durch einen Grünstreifen von der Fahrbahn getrennt und repräsentiert daher einen nicht-befahrbaren OpenDRIVE-Fahrstreifen.

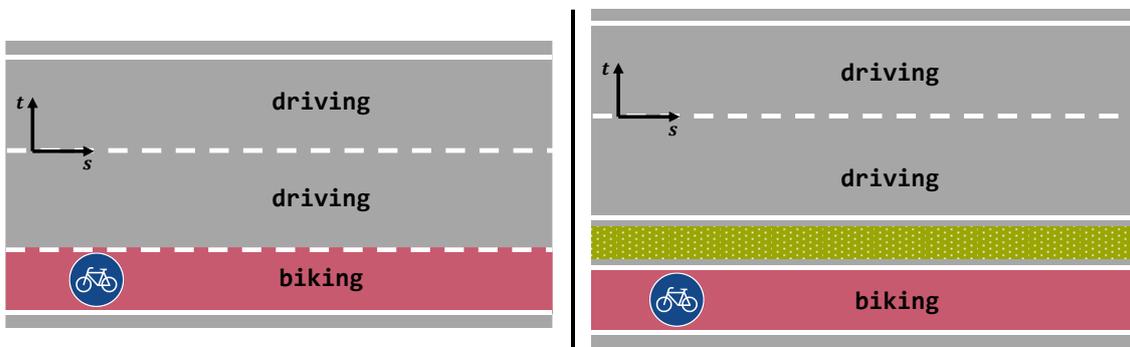


Abbildung 6-5: Beispiel befahrbarer und nicht-befahrbarer Fahrradweg

Als Resultat des Algorithmus A.1 ist jeder OpenDRIVE-Fahrstreifen als befahrbar bzw. nicht-befahrbar klassifiziert. Dies wird in einer Datenstruktur gespeichert, die für jeden OpenDRIVE-Fahrstreifen einen Eintrag beinhaltet, der die Klassifizierung dieses OpenDRIVE-Fahrstreifens als befahrbar bzw. nicht-befahrbar enthält. Diese Datenstruktur wird im Schaubild der Implementierung des zweiten Konzept-Schritts (siehe Abbildung 6-4) als „**Befahrbarkeit OpenDRIVE-Fahrstreifen**“ bezeichnet.

### 6.3.2.2 Algorithmus A.2: Manuelles Bearbeiten der befahrbaren Fahrstreifen

Der zweite Teil des zweiten Konzept-Schritts ist die Durchführung des Algorithmus A.2 „Manuelles Bearbeiten der befahrbaren Fahrstreifen“. Es ist in Einzelfällen möglich, dass die automatisierte

Klassifizierung eines OpenDRIVE-Fahrstreifens als befahrbar bzw. nicht-befahrbar anhand des `type`-Attributs zu einer falschen Klassifizierung führt (siehe Abschnitt 5.3.2.2). Aufgrund dessen bietet dieser Algorithmus die Möglichkeit, die vom Algorithmus A.1 vorgenommene Einteilung der Befahrbarkeit der OpenDRIVE-Fahrstreifen manuell zu bearbeiten. Die Ausführung des Algorithmus A.2 ist optional und wird per Nutzereingabe gestartet. Falls die Ausführung des Algorithmus erwünscht ist, wird durch jede Straße der OpenDRIVE-Karte iteriert. In jeder Straße ist es einerseits möglich, die vom Algorithmus A.1 als nicht-befahrbar klassifizierten OpenDRIVE-Fahrstreifen als befahrbar zu klassifizieren (Hinzufügen von befahrbaren OpenDRIVE-Fahrstreifen) sowie andererseits die als befahrbar klassifizierten OpenDRIVE-Fahrstreifen als nicht-befahrbar zu klassifizieren (Entfernen von befahrbaren OpenDRIVE-Fahrstreifen).

Für das Hinzufügen von befahrbaren OpenDRIVE-Fahrstreifen erfolgt für jede Straße eine Ausgabe von zwei Tabellen. Die erste Tabelle listet die vom Algorithmus A.1 als befahrbar klassifizierten OpenDRIVE-Fahrstreifen, während die zweite Tabelle alle als nicht-befahrbar klassifizierten OpenDRIVE-Fahrstreifen zeigt. Dem Nutzer wird anschließend die Möglichkeit gegeben, einen bzw. mehrere nicht-befahrbare OpenDRIVE-Fahrstreifen als befahrbar zu klassifizieren. Dies erfolgt durch Eingabe des Index des entsprechenden OpenDRIVE-Fahrstreifens bzw. der Indizes der entsprechenden OpenDRIVE-Fahrstreifen aus der angezeigten Tabelle. Wird ein gültiger Index eingegeben bzw. werden mehrere gültige Indizes eingegeben, werden die Einträge der entsprechenden OpenDRIVE-Fahrstreifen in der Datenstruktur „Befahrbarkeit OpenDRIVE-Fahrstreifen“ auf befahrbar geändert. Abbildung 6-6 zeigt eine Konsole bei einer beispielhaften Ausführung des Algorithmus A.2 vor einer Nutzereingabe und nach der Nutzereingabe. Es erfolgt die Eingabe "0, 1" (links dargestellte Konsole), was dazu führt, dass die OpenDRIVE-Fahrstreifen mit der `id` -2 und -3 (*lane section* 0.0) als befahrbar klassifiziert werden (siehe rechts dargestellte Konsole).

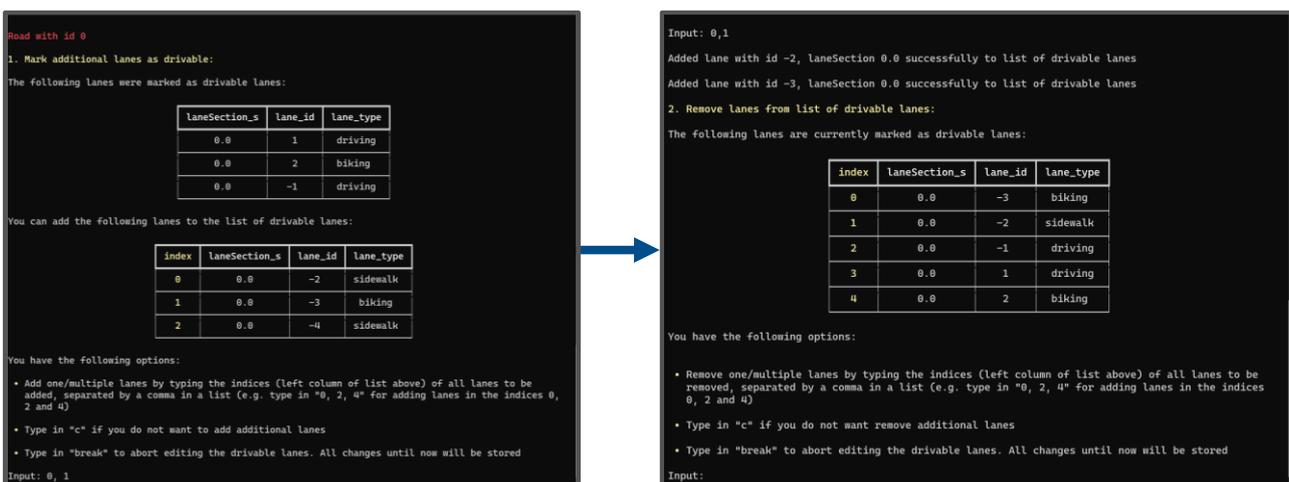


Abbildung 6-6: Konsole bei Ausführung des Algorithmus A.2

Nach dem Hinzufügen von befahrbaren OpenDRIVE-Fahrstreifen wird dem Nutzer die Möglichkeit des Entfernens von befahrbaren OpenDRIVE-Fahrstreifen geboten (siehe rechte Konsole, Abbildung 6-6). Die Funktionsweise ist dabei analog zum Hinzufügen von befahrbaren OpenDRIVE-Fahrstreifen. Nach erfolgreichem Hinzufügen und Entfernen von befahrbaren OpenDRIVE-Fahrstreifen in einer Straße wird zur nächsten Straße übergegangen. Sobald durch alle Straßen iteriert wurde bzw. sobald

der Nutzer die manuelle Eingabe vorzeitig beendet hat, wird die durch die manuellen Eingaben modifizierte Datenstruktur „Befahrbarkeit OpenDRIVE-Fahrstreifen“ zurückgegeben. Die Durchführung des Algorithmus A.4 „Manuelles Bearbeiten der Segmente“ ist abgeschlossen.

### 6.3.2.3 Modifikation des OpenDRIVE-ElementTree

Im dritten Teil des zweiten Konzept-Schritts wird im OpenDRIVE-*ElementTree* in jeder Straße, die mindestens einen befahrbaren OpenDRIVE-Fahrstreifen besitzt, unterhalb jedes `<lanes>`-Elements das BSSD-Wurzelement (Element `<userData>`) erstellt. Das Element `<userData>` besitzt das verpflichtende Attribut `code` sowie das optionale Attribut `value` (siehe Abschnitt 3.3.3). Um eine Unterscheidung der BSSD-Wurzelemente zu anderen in der OpenDRIVE-Datei möglicherweise vorhandenen `<userData>`-Elementen zu ermöglichen, werden den Attributen die Werte `code="BSSD"` und `value="BSSD_segments"` zugewiesen.

### 6.3.3 Schritt 3: Erstellen der BSSD-Segmente

Im dritten Konzept-Schritt „Erstellen der BSSD-Segmente“ werden in jeder Straße der OpenDRIVE-Karte, die mindestens einen befahrbaren OpenDRIVE-Fahrstreifen besitzt, die Segmente der BSSD erstellt (siehe Abschnitt 5.3.1.3). Das Funktionsschaubild des dritten Konzept-Schritts ist in Abbildung 6-7 dargestellt, die den zu diesem Schritt zugehörigen Teil des Gesamtschaubilds der Implementierung beinhaltet.

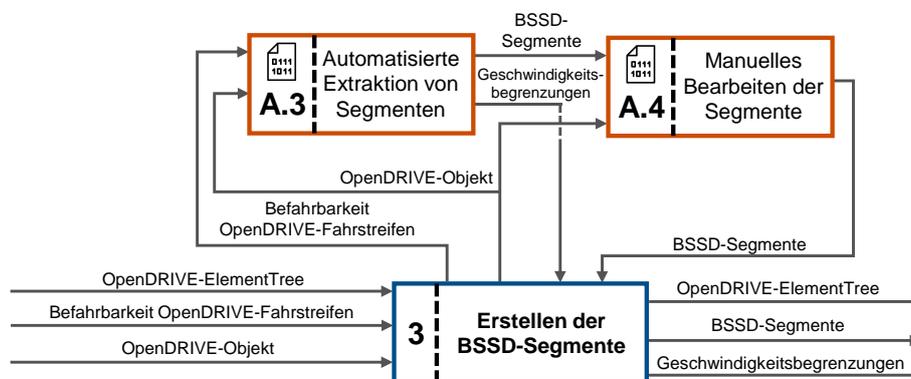


Abbildung 6-7: Funktionsschaubild Schritt 3 „Einlesen & Prüfen der OpenDRIVE-Datei“

Die Implementierung des dritten Konzept-Schritts besteht aus drei aufeinanderfolgenden Teilen, die im Folgenden näher erläutert werden.

#### 6.3.3.1 Algorithmus A.3: Automatisierte Extraktion von Segmenten

Der erste Teil des dritten Konzept-Schritts ist die Durchführung des Algorithmus A.3 „Automatisierte Extraktion von Segmenten“. Bei diesem Algorithmus werden in jeder Straße, basierend auf den in der OpenDRIVE-Karte enthaltenen BSSD-relevanten Informationen, *s*-Koordinaten extrahiert, bei denen der Definitionsbeginn bzw. das Definitionsende eines BSSD-Segments liegt. Die automatisierte Extraktion der BSSD-Segmente basiert auf Regeln, deren Erfüllung den Definitionsbeginn bzw. das Definitionsende eines BSSD-Segments zur Folge haben. Im Rahmen dieser Arbeit werden

---

sieben Regeln für das automatisierte Erkennen des Definitionsbeginns sowie eine Regel für das automatisierte Erkennen des Definitionsendes eines BSSD-Segments definiert. Im Folgenden werden diese näher erläutert.

In jedem Bereich der *s*-Koordinate, der befahrbare OpenDRIVE-Fahrstreifen beinhaltet, ist erforderlich, dass ein BSSD-Segment definiert ist. Daraus ergibt sich

**Regel 1: Bei dem Definitionsbeginn der ersten *lane section* einer Straße, die befahrbare OpenDRIVE-Fahrstreifen beinhaltet, ist die Definition eines BSSD-Segments erforderlich.**

Jeder befahrbare OpenDRIVE-Fahrstreifen besitzt mindestens einen äquivalenten BSSD-Fahrstreifen. Somit ist bei jedem neu definierten befahrbaren OpenDRIVE-Fahrstreifen (besitzt keinen vorangehenden OpenDRIVE-Fahrstreifen) die Definition eines neuen BSSD-Fahrstreifens und somit die Definition eines neuen BSSD-Segments erforderlich (siehe beispielweise Abbildung 5-10, Segment B). Analog ist bei jedem Definitionsende eines OpenDRIVE-Fahrstreifens (besitzt keine nachfolgenden OpenDRIVE-Fahrstreifen) die Definition des Endes eines bestehenden BSSD-Fahrstreifens notwendig. Auch in diesem Fall ist die Definition eines neuen BSSD-Segments erforderlich. Aus diesen beiden Tatsachen resultiert

**Regel 2: Bei dem Definitionsbeginn jeder *lane section*, deren Gesamtanzahl der befahrbaren OpenDRIVE-Fahrstreifen sich im Vergleich zur vorangehenden *lane section* ändert, ist die Definition eines BSSD-Segments erforderlich.**

Es ist möglich, dass beim Übergang von einer *lane section* zu einer anderen *lane section* gleichzeitig ein befahrbarer OpenDRIVE-Fahrstreifen endet und ein befahrbarer OpenDRIVE-Fahrstreifen beginnt (siehe beispielweise Abbildung A-4, Segment 99.46). Dieser Sonderfall wird von Regel 2 nicht abgedeckt, da sich dabei die Gesamtanzahl der befahrbaren OpenDRIVE-Fahrstreifen nicht ändert. Um diesen Sonderfall abzudecken, werden

**Regel 3: Bei dem Definitionsbeginn jeder *lane section*, die eine vorangehende *lane section* besitzt, und einen befahrbaren OpenDRIVE-Fahrstreifen beinhaltet, der keinen vorangehenden OpenDRIVE-Fahrstreifen besitzt (kein Verweis im Element <predecessor>), ist die Definition eines BSSD-Segments erforderlich**

und

**Regel 4: Bei dem Definitionsende jeder *lane section*, die eine nachfolgende *lane section* mit mindestens einem befahrbaren OpenDRIVE-Fahrstreifen besitzt, und einen befahrbaren OpenDRIVE-Fahrstreifen beinhaltet, der keinen nachfolgenden OpenDRIVE-Fahrstreifen besitzt (kein Verweis im Element <successor>), ist die Definition eines BSSD-Segments erforderlich**

definiert.

Auf Basis der Anzahl der befahrbaren OpenDRIVE-Fahrstreifen einer *lane section* ist es darüber hinaus möglich, eventuell vorhandene BSSD-Definitionslücken einer Straße zu finden und somit das

---

Definitionsende eines BSSD-Segments festzulegen (siehe beispielweise Abbildung 5-9). Daraus ergibt sich

**Regel 5: Bei dem Definitionsende jeder *lane section*, die mindestens einen befahrbaren OpenDRIVE-Fahrstreifen besitzt, und die eine nachfolgende *lane section* ohne befahrbaren OpenDRIVE-Fahrstreifen besitzt, muss das Definitionsende eines BSSD-Segments definiert werden**

für das Finden von BSSD-Definitionslücken.

In OpenDRIVE besteht die Möglichkeit der Definition der in einem OpenDRIVE-Fahrstreifen geltenden zulässigen Höchstgeschwindigkeit (Element `<speed>`, siehe Abschnitt 3.3.1.3). Da diese dem BSSD-Verhaltensattribut *Speed Limit* entlang der vorgesehenen Fahrtrichtung in dem äquivalenten BSSD-Fahrstreifen entspricht, ist die Extraktion von BSSD-Segmenten basierend auf dem `<speed>`-Element möglich. Es ergibt sich

**Regel 6: Bei dem Definitionsbeginn jedes `<speed>`-Elements, das in einem befahrbaren OpenDRIVE-Fahrstreifen definiert ist und einen Wechsel der in dem OpenDRIVE-Fahrstreifen zulässigen Höchstgeschwindigkeit bedingt, ist die Definition eines BSSD-Segments erforderlich.**

Weiterhin bietet OpenDRIVE die Möglichkeit, Verkehrszeichen zu modellieren (statische `<signal>`- und `<signalReference>`-Elemente, siehe Abschnitt 3.3.1.4 und 3.3.1.5). Je nach der durch das Verkehrszeichen definierten Regel ist es möglich, dass ein BSSD-Verhaltensattribut durch dieses Verkehrszeichen beeinflusst wird oder nicht. Im Rahmen dieser Arbeit wird die BSSD im Kontext der StVO betrachtet. Somit ist es möglich, für die einzelnen Verkehrszeichen des VzKat zu analysieren, ob diese ein im Kontext der BSSD relevantes Indikationselement repräsentieren oder nicht. Diese Analyse wird von Glatzki et al.<sup>79</sup> durchgeführt und in dieser Arbeit verwendet. Daraus ergibt sich

**Regel 7: Bei dem Gültigkeitsbeginn jedes statischen `<signal>`- bzw. `<signalReference>`-Elements, das entsprechend des VzKat ein in Deutschland gültiges Verkehrszeichen repräsentiert, ein Indikationselement darstellt und dessen Gültigkeitsbeginn bei einer *s*-Koordinate definiert ist, bei der mindestens ein befahrbarer OpenDRIVE-Fahrstreifen existiert, wird ein BSSD-Segment definiert.**

Die Erfüllung von Regel 7 bietet im Vergleich zu den vorigen Regeln keine absolute Sicherheit, dass die Definition eines neuen BSSD-Segments erforderlich ist. Die Ursache dafür ist die Verwendung von Verkehrszeichen im Straßenverkehr, die eine bereits bestehende Regel wiederholen und somit keine Veränderung eines BSSD-Verhaltensattributs zur Folge haben. Beispielsweise auf Autobahnen wird das Verkehrszeichen zulässige Höchstgeschwindigkeit (Zeichen 274, VzKat) wiederholt aufgestellt. Für eine korrekte Segmentierung entsprechend der in der OpenDRIVE-Karte vorhandenen Verkehrszeichen wäre erforderlich, zusätzlich zum Gültigkeitsbeginn der Verkehrszeichen, das Gültigkeitsende der Verkehrszeichen zu kennen. Dies würde eine vollständige Modellierung der

---

<sup>79</sup> Glatzki, F.; Winner, H.: Inferenz von Verhaltensattributen der verhaltenssemantischen Szeneriebeschreibung (2022), S. 4.

---

Verkehrsregeln entsprechend der StVO erfordern, die den Rahmen dieser Arbeit übersteigt. Aufgrund dessen und da davon auszugehen ist, dass der Sonderfall von wiederholenden Verkehrszeichen selten auftritt, wird die aus dieser Regel resultierende Ungenauigkeit der Segmentierung im Rahmen dieser Arbeit akzeptiert.

Neben der Modellierung von Verkehrszeichen ist es in OpenDRIVE möglich, Lichtsignalanlagen und weitere dynamische Signalelemente des Straßenverkehrs zu modellieren (dynamische `<signal>`- und `<signalReference>`-Elemente, siehe Abschnitt 3.3.1.4 und 3.3.1.5). Da davon auszugehen ist, dass diese im Allgemeinen ebenfalls eine Beeinflussung mindestens eines BSSD-Verhaltensattributs zur Folge haben, wird

**Regel 8: Bei dem Gültigkeitsbeginn jedes dynamischen `<signal>`- bzw. `<signalReference>`-Elements, dessen Gültigkeitsbeginn bei einer *s*-Koordinate definiert ist, bei der mindestens ein befahrbarer OpenDRIVE-Fahrstreifen existiert, wird ein BSSD-Segment definiert**

festgelegt. Analog zu Regel 7 bietet die Erfüllung von Regel 8 keine absolute Sicherheit, dass die Definition eines neuen BSSD-Segments erforderlich ist. Die Ursache dafür ist, dass in der Spezifikation von OpenDRIVE kein Attribut vorgesehen ist, das die reale Repräsentation eines dynamischen `<signal>`-Elements definiert. Da dynamische Elemente des Straßenverkehrs in der Regel nicht standardisiert dokumentiert sind (siehe beispielsweise VzKat für Verkehrszeichen), werden die Attribute `type` und `subtype` bei dynamischen `<signal>`-Elementen im Gegensatz zu statischen `<signal>`-Elementen im Allgemeinen nicht verwendet. Somit ist eine eindeutige Identifikation des realen Elements, das durch das dynamische `<signal>`-Element repräsentiert wird, nicht direkt möglich. Grundsätzlich besteht die Möglichkeit, die Repräsentation eines dynamischen `<signal>`-Elements durch das optionale Attribut `name` festzulegen, das einen beliebigen Freitext beinhaltet. Da die Verwendung dieses Attributs jedoch optional ist und der Inhalt frei wählbar ist, ist eine automatisierte Bestimmung der realen Repräsentation eines dynamischen `<signal>`-Elements (z.B. Ampel, Schranke) im Allgemeinen nicht möglich. Somit ist es möglich, dass OpenDRIVE-Karten existieren, die dynamische `<signal>`-Elemente beinhalten, deren reale Repräsentation kein BSSD-Verhaltensattribut beeinflusst. Da jedoch davon auszugehen ist, dass der Großteil der dynamischen `<signal>`-Elemente eine Beeinflussung mindestens eines BSSD-Verhaltensattributs zur Folge hat, wird diese Unsicherheit im Rahmen dieser Arbeit akzeptiert.

Alle in diesem Abschnitt definierten Regeln für das automatisierte Erkennen des Definitionsbeginns bzw. -endes eines BSSD-Segments sind zusammenfassend im Anhang A.4, Tabelle A-6 gelistet.

Die Implementierung der definierten Regeln erfolgt in fünf Skripten, die inkrementell vom Algorithmus A.1 („Automatisierte Extraktion von Segmenten“) aufgerufen werden. Das Resultat von jedem dieser Skripte sind *s*-Koordinaten, bei denen ein BSSD-Segment beginnt bzw. endet. Zur Speicherung dieser *s*-Koordinaten wird eine Datenstruktur erstellt, die für jedes zu definierende BSSD-Segment einen separaten Eintrag enthält. Jeder Eintrag enthält als Werte die *id* der Straße in der das BSSD-Segment zu erstellen ist, die *s*-Koordinate, bei der der Definitionsbeginn des BSSD-Segments ist, sowie, falls auf das Segment eine BSSD-Definitionslücke folgt, die *s*-Koordinate, bei der das

---

Definitionsende des BSSD-Segments ist. Diese Datenstruktur wird im Schaubild der Implementierung des dritten Konzept-Schritts (siehe Abbildung 6-7) als „**BSSD-Segmente**“ bezeichnet. Jedes der fünf Skripte fügt die gefundenen BSSD-Segmente in die beschriebene Datenstruktur ein und übergibt diese am Ende der Ausführung des Skripts an das nachfolgende Skript. Somit sind alle automatisiert gefundenen BSSD-Segmente nach Ausführung der fünf Skripte in der Datenstruktur „BSSD-Segmente“ enthalten. Die Funktionsweise der einzelnen Skripte wird im Folgenden näher beschrieben.

### **1. Segmente basierend auf der Anzahl der befahrbaren Fahrstreifen**

Dieses Skript setzt die Regeln 1, 2 und 5 für die automatisierte Extraktion von BSSD-Segmenten um. Durch die Einteilung aller OpenDRIVE-Fahrstreifen in befahrbare und nicht-befahrbare OpenDRIVE-Fahrstreifen (siehe Datenstruktur „Befahrbarkeit OpenDRIVE-Fahrstreifen“, Abschnitt 6.3.2) ist es in jeder Straße möglich, für jede *lane section* die Anzahl der befahrbaren und nicht-befahrbaren OpenDRIVE-Fahrstreifen zu berechnen. Basierend auf dieser Information wird die Erfüllung der Regeln 1, 2 und 5 für jede *lane section* überprüft. Bei Erfüllung einer Regel durch eine *lane section* wird folglich der Beginn eines BSSD-Segments beim Definitionsbeginn (Regel 1 und 2) bzw. das Ende eines BSSD-Segments beim Definitionsende (Regel 5) der *lane section* definiert.

### **2. Segmente basierend auf dem Beginn/Ende eines befahrbaren OpenDRIVE-Fahrstreifens**

Dieses Skript setzt die Regeln 3 und 4 für die automatisierte Extraktion von BSSD-Segmenten um. Dazu wird in jeder Straße durch alle befahrbaren OpenDRIVE-Fahrstreifen iteriert. Anhand der Betrachtung des OpenDRIVE-Objekts, das die Informationen über den vorangehenden bzw. nachfolgenden OpenDRIVE-Fahrstreifen beinhaltet, wird für jeden befahrbaren OpenDRIVE-Fahrstreifen geprüft, ob ein vorangehender bzw. nachfolgender OpenDRIVE-Fahrstreifen existiert. Wenn dies nicht der Fall ist, wird anhand der Datenstruktur „Befahrbarkeit OpenDRIVE-Fahrstreifen“ geprüft, ob eine vorangehende bzw. nachfolgende *lane section* mit befahrbaren OpenDRIVE-Fahrstreifen existiert. Bei Erfüllung der beiden genannten Bedingungen wird der Beginn eines BSSD-Segments beim Definitionsbeginn (Regel 3) bzw. Definitionsende (Regel 4) des betrachteten OpenDRIVE-Fahrstreifens definiert.

### **3. Segmente basierend auf Geschwindigkeitsbegrenzungen**

Dieses Skript setzt Regel 6 für die automatisierte Extraktion von BSSD-Segmenten um. Dazu wird in jedem befahrbaren OpenDRIVE-Fahrstreifen durch alle <speed>-Elemente iteriert. Dabei wird eine Datenstruktur erstellt, die für jedes <speed>-Element, das in einem befahrbaren OpenDRIVE-Fahrstreifen definiert ist, einen Eintrag enthält, der die Informationen dieses <speed>-Elements (*id* des OpenDRIVE-Fahrstreifens, zulässige Höchstgeschwindigkeit, *s*-Position des Gültigkeitsbeginns) speichert. Diese Datenstruktur wird im Schaubild der Implementierung des dritten Konzept-Schritts (siehe Abbildung 6-7) als „**Geschwindigkeitsbegrenzungen**“ bezeichnet. Anhand dieser Datenstruktur wird für jedes <speed>-Element eines befahrbaren OpenDRIVE-Fahrstreifens geprüft, ob dieses einen Wechsel der zulässigen Höchstgeschwindigkeit bedingt. Dazu ist erforderlich, die zu diesem OpenDRIVE-Fahrstreifen zugehörige zuvor definierte zulässige Höchstgeschwindigkeit zu finden, um diese mit der zulässigen Höchstgeschwindigkeit im betrachteten <speed>-Element zu vergleichen. Die zuvor gültige zulässige Höchstgeschwindigkeit ist entweder im gleichen

---

OpenDRIVE-Fahrstreifen (Niedrigerer Wert des Attributs `sOffset`), im vorangehenden OpenDRIVE-Fahrstreifen oder gar nicht definiert. Bei einer Veränderung der zulässigen Höchstgeschwindigkeit durch das betrachtete `<speed>`-Element wird bei der `s`-Koordinate dieses Elements der Beginn eines BSSD-Segments definiert.

#### 4. Segmente basierend auf Verkehrszeichen

Dieses Skript setzt Regel 7 für die automatisierte Extraktion von BSSD-Segmenten um. Dazu wird in jeder Straße durch alle statischen `<signal>`-Elemente sowie alle `<signalReference>`-Elemente, die auf ein statisches `<signal>`-Element verweisen, iteriert. Bei jedem dieser Elemente wird zunächst überprüft, ob dieses ein in Deutschland gültiges Verkehrszeichen entsprechend des VzKat repräsentiert. Dazu wird die Ausprägung des Attributs `country` des `<signal>`-Elements betrachtet (siehe Abschnitt 3.3.1.4). Sofern das Element ein in Deutschland gültiges Verkehrszeichen repräsentiert, wird überprüft, ob das Verkehrszeichen ein Indikationselement darstellt. Dazu werden die Attribute `type` und `subtype` des `<signal>`-Elements betrachtet (siehe Abschnitt 3.3.1.4). Entsprechend der Klassifizierung von Glatzki et al.<sup>80</sup> wird das `<signal>`-Element als Indikationselement klassifiziert oder nicht. Sofern die beiden genannten Bedingungen erfüllt sind, wird zuletzt geprüft, ob bei der `s`-Koordinate, bei der das Verkehrszeichen Gültigkeit annimmt (Attribut `s`), mindestens ein befahrbarer OpenDRIVE-Fahrstreifen definiert ist. Bei Erfüllung der genannten Bedingungen wird ein BSSD-Segment bei der `s`-Koordinate des statischen `<signal>`- bzw. `<signalReference>`-Elements definiert.

#### 5. Segmente basierend auf dynamischen Elementen des Straßenverkehrs

Dieses Skript setzt Regel 8 für die automatisierte Extraktion von BSSD-Segmenten um. Analog zur Funktionsweise des vorigen Skripts („Segmente basierend auf Verkehrszeichen“) wird durch alle dynamischen `<signal>`- bzw. `<signalReference>`-Elemente iteriert und anhand des Attributs `s` die Erfüllung der in Regel 8 definierten Bedingung geprüft. Beim Erfüllen der Bedingung wird folglich ein BSSD-Segment bei der `s`-Koordinate des dynamischen `<signal>`- bzw. `<signalReference>`-Elements definiert.

Nach Durchführung der fünf Skripte ist die Ausführung des Algorithmus A.3 beendet. Dazu werden die Datenstrukturen „BSSD-Segmente“ und „Geschwindigkeitsbegrenzungen“ zurückgegeben.

#### 6.3.3.2 Algorithmus A.4: Manuelles Bearbeiten der Segmente

Der zweite Teil des dritten Konzept-Schritts ist die Durchführung des Algorithmus A.4 „Manuelles Bearbeiten der Segmente“. Es ist davon auszugehen, dass anhand der im Algorithmus A.3 definierten Regeln einerseits nicht alle BSSD-Segmente gefunden werden (siehe Abschnitt 5.3.2.3) sowie andererseits falsche BSSD-Segmente gefunden werden, da die Erfüllung der Regeln 7 und 8 keine absolute Sicherheit für die korrekte Definition eines BSSD-Segments garantiert (siehe Abschnitt 6.3.3.1). Aufgrund dessen bietet dieser Algorithmus die Möglichkeit der Definition von weiteren BSSD-

---

<sup>80</sup> Glatzki, F.; Winner, H.: Inferenz von Verhaltensattributen der verhaltenssemantischen Szeneriebeschreibung (2022), S. 4.

Segmenten (Hinzufügen von BSSD-Segmenten) sowie des Löschens von automatisch gefundenen BSSD-Segmenten (Entfernen von BSSD-Segmenten). Die Ausführung des Algorithmus ist optional und wird per Nutzereingabe gestartet.

Falls die Ausführung des Algorithmus erwünscht ist, wird durch jede Straße der OpenDRIVE-Karte iteriert, die mindestens einen befahrbaren OpenDRIVE-Fahrstreifen beinhaltet. Dabei ist es in jeder dieser Straßen zunächst möglich, neue BSSD-Segmente hinzuzufügen. Dazu erfolgt eine Ausgabe der bereits in dieser Straße definierten BSSD-Segmente (siehe Datenstruktur „BSSD-Segmente“). Dem Nutzer wird anschließend die Möglichkeit gegeben, eine *s*-Koordinate einzugeben, bei der ein weiteres BSSD-Segment in der aktuellen Straße erstellt wird. Da bei Anwendung von Regel 5 (Algorithmus A.3) alle BSSD-Definitionslücken der vorhandenen OpenDRIVE-Karte automatisch gefunden werden, ist nur die Eingabe der *s*-Koordinate des Definitionsbeginns des BSSD-Segments erforderlich. Anhand des OpenDRIVE-Objekts und der bisher erstellten BSSD-Segmente wird geprüft, ob bei der eingegebenen *s*-Koordinate die Definition eines BSSD-Segments möglich ist. Ein Beispiel für ein ungültige Eingabe ist eine *s*-Koordinate, bei der eine BSSD-Definitionslücke existiert oder eine *s*-Koordinate, welche die Länge der Straße übersteigt. Bei Eingabe einer gültigen *s*-Koordinate wird der Datenstruktur „BSSD-Segmente“ ein BSSD-Segment mit dieser *s*-Koordinate hinzugefügt. Abbildung 6-8 zeigt eine Konsole bei einer beispielhaften Ausführung des Algorithmus A.4 vor und nach einer Nutzereingabe. Es erfolgt die Eingabe einer *s*-Koordinate von "50" (links dargestellte Konsole), was zu der Definition eines BSSD-Segments bei dieser *s*-Koordinate führt (siehe rechts dargestellte Konsole). In dem dargestellten Beispiel existiert eine BSSD-Definitionslücke bei einer *s*-Koordinate von "75.0". Diese wird von dem bereits existierenden BSSD-Segment mit *sStart*="25.0" übernommen, sodass das Attribut *sEnd* des erstellten BSSD-Segments mit dem Wert "75.0" belegt wird.

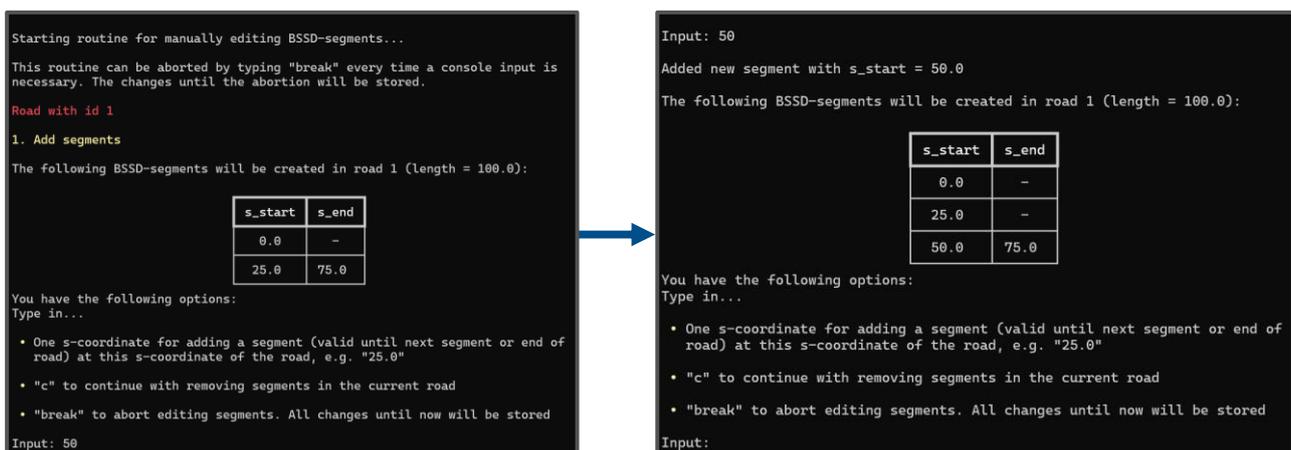


Abbildung 6-8: Konsole bei Ausführung des Algorithmus A.4

Nach erfolgter Eingabe einer *s*-Koordinate hat der Nutzer die Möglichkeit, weitere BSSD-Segmente in der aktuellen Straße zu erstellen oder zum Entfernen von BSSD-Segmenten in dieser Straße überzugehen. Beim Entfernen von BSSD-Segmenten erfolgt analog zum Hinzufügen von BSSD-Segmenten die Ausgabe der bereits in dieser Straße definierten BSSD-Segmente. Durch die Eingabe der *s*-

---

Koordinate des zu entfernenden BSSD-Segments wird dem Nutzer die Möglichkeit geboten, ein bestehendes BSSD-Segment zu löschen.

Nach erfolgreichem Hinzufügen und Entfernen von BSSD-Segmenten in einer Straße wird zur nächsten Straße übergegangen. Sobald durch alle Straßen mit mindestens einem befahrbaren OpenDRIVE-Fahstreifen iteriert wurde bzw. sobald der Nutzer die manuelle Eingabe vorzeitig beendet hat, wird die um die manuell hinzugefügten bzw. gelöschten BSSD-Segmente erweiterte Datenstruktur „BSSD-Segmente“ zurückgegeben. Die Durchführung des Algorithmus A.4 „Manuelles Bearbeiten der Segmente“ ist abgeschlossen.

### 6.3.3.3 Modifikation des OpenDRIVE-ElementTree

Im dritten Teil des dritten Konzept-Schritts wird der *OpenDRIVE-ElementTree* um die in den beiden vorigen Algorithmen A.3 und A.4 automatisiert gefundenen bzw. manuell erstellten BSSD-Segmente erweitert. Dazu wird durch jedes `<road>`-Element iteriert, das ein BSSD-Wurzelement enthält und anhand des `id`-Attributs des `<road>`-Elements werden alle Einträge der Datenstruktur „BSSD-Segmente“ für die aktuelle Straße gefiltert. Für jeden der gefilterten Einträge wird ein BSSD-Segment in Form eines Elements `<segment>` erstellt. Letzteres besitzt das verpflichtende Attribut `sStart` entsprechend des Definitionsbeginns des BSSD-Segments und das optionale Attribut `sEnd`, das im Fall einer auf das BSSD-Segment folgenden BSSD-Definitionslücke definiert wird. Dabei werden die Zahlen, wie alle im Rahmen der BSSD-Integration eingefügten Gleitkommazahlen, entsprechend der Norm IEEE 754<sup>81</sup> mit 17 Signifikaten Stellen gespeichert.

### 6.3.4 Schritt 4: Erstellen der Fahstreifen-Gruppierungen

Im vierten Konzept-Schritt „Erstellen der Fahstreifen-Gruppierungen“ werden im *OpenDRIVE-ElementTree* unterhalb jedes im vorigen Konzept-Schritt erstellten `<segment>`-Elements ein Element `<right>` und ein Element `<left>` erstellt. Diese Elemente dienen der Gruppierung der BSSD-Fahstreifen eines BSSD-Segments in die rechte und linke Straßenseite, was der Verbesserung der menschlichen Lesbarkeit der OpenDRIVE-Datei dient.

### 6.3.5 Schritt 5: Erstellen der BSSD-Fahstreifen

Im fünften Konzept-Schritt „Erstellen der BSSD-Fahstreifen“ werden in jedem zuvor erstellten BSSD-Segment die BSSD-Fahstreifen basierend auf den befahrbaren OpenDRIVE-Fahstreifen erstellt (siehe Abschnitt 5.3.1.5). Das Funktionsschaubild des fünften Konzept-Schritts ist in Abbildung 6-9 dargestellt, die den zu diesem Schritt zugehörigen Teil des Gesamtschaubilds der Implementierung beinhaltet. Die Implementierung des fünften Konzept-Schritts besteht aus zwei aufeinanderfolgenden Teilen, die im Folgenden näher erläutert werden.

---

<sup>81</sup> IEEE: IEEE Standard for Floating-Point Arithmetic (2019).

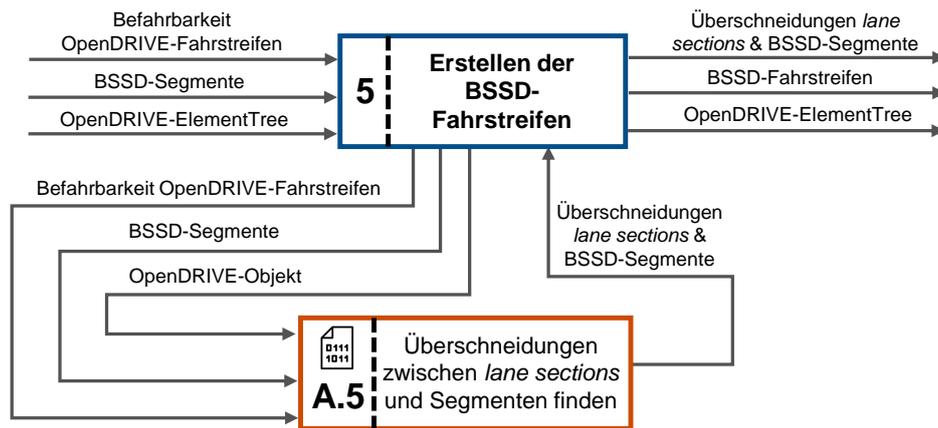


Abbildung 6-9: Funktionsschaubild Schritt 5 „Erstellen der BSSD-Fahrstreifen“

### 6.3.5.1 Algorithmus A.5: Überschneidungen zwischen lane sections und Segmenten finden

Der erste Teil des fünften Konzept-Schritts ist die Durchführung des Algorithmus A.5 „Überschneidungen zwischen *lane sections* und Segmenten finden“. Bei diesem Algorithmus wird durch jedes BSSD-Segment iteriert, wobei jeweils alle *lane sections* gesucht werden, deren Definitionsbereich der *s*-Koordinate mit dem Definitionsbereich der *s*-Koordinate dieses BSSD-Segments überschneiden. Darauf basierend wird eine Datenstruktur erstellt, die je Überschneidung zwischen einem BSSD-Segment und einer *lane section* einen Eintrag enthält. Diese Datenstruktur wird im Schaubild der Implementierung des fünften Konzept-Schritts (siehe Abbildung 6-9) als „**Überschneidungen lane sections & BSSD-Segmente**“ bezeichnet.

Es sei auf die Beachtung des Spezialfalls von einseitig definierten *lane sections* (Attribut `singleSide="true"`) im Kontext des Algorithmus A.5 hingewiesen. Dadurch ist es möglich, dass in einem longitudinalen Teilabschnitt einer Straße zwei *lane sections* definiert sind. Dieser Sonderfall bedarf einer speziellen Betrachtung im Kontext der Überschneidungen zwischen *lane sections* und BSSD-Segmenten. Bei einseitig definierten *lane sections* ist es möglich, dass für bestimmte *s*-Koordinaten eines BSSD-Segments zwei *lane sections* gleichzeitig zu diesem BSSD-Segment überschneiden. Die praktische Verwendung von einseitig definierten *lane sections* wird in der Spezifikation von OpenDRIVE<sup>82</sup> nur bedingt beschrieben, sodass Fragen bezüglich der konkreten Modellierung von einseitig definierten *lane sections* offenbleiben. Auch durch eine Anfrage bei ASAM war es nicht möglich, diese Fragen vollständig zu klären. Da es weiterhin nicht möglich war, beispielhafte OpenDRIVE-Karten, die einseitig definierte *lane sections* verwenden, zu finden, ist davon auszugehen, dass dieser Sonderfall in der Praxis nicht relevant ist. Daraus ergibt sich auch, dass falls OpenDRIVE-Karten mit einseitig definierten *lane sections* existieren, die Definition und Verwendung dieser *lane sections* wahrscheinlich nicht einheitlich erfolgt. Darüber hinaus unterstützt das für die Implementierung dieser Arbeit verwendete Python-Modul `opendrive_parser` keine einseitig definierten *lane sections*. Aus den genannten Gründen wird darauf verzichtet, diesen Sonderfall bei der zu dieser Arbeit zugehörigen Implementierung zu berücksichtigen.

<sup>82</sup> ASAM e.V.: Specification ASAM OpenDRIVE, Version 1.7.0 (2021), Kap. 9.2.

### 6.3.5.2 Modifikation des OpenDRIVE-ElementTree

Im zweiten Teil des fünften Konzept-Schritts werden die BSSD-Fahrstreifen im OpenDRIVE-*ElementTree* unterhalb der BSSD-Segmente erstellt. Entsprechend der bei der Konzeptdetaillierung in Abschnitt 5.3.1.5 definierten Regel werden die BSSD-Fahrstreifen in einem BSSD-Segment anhand der befahrbaren OpenDRIVE-Fahrstreifen der ersten *lane section*, die mit diesem BSSD-Segment überschneidet, definiert (siehe beispielsweise Abbildung 5-10 und Tabelle 5-7). Aufgrund dessen wird durch jedes erstellte BSSD-Segment iteriert und anhand der zuvor erstellten Datenstruktur „Überschneidungen *lane sections* & BSSD-Segmente“ die beim Definitionsbeginn dieses BSSD-Segments definierte *lane section* gefunden. Für jeden befahrbaren OpenDRIVE-Fahrstreifen in dieser *lane section* wird ein äquivalenter BSSD-Fahrstreifen (Element `<lane>`) erstellt. Dazu wird die Ausprägung des Attributs `id` von diesem befahrbaren OpenDRIVE-Fahrstreifen übernommen. Das `<lane>`-Element wird je nach Vorzeichen der `id` unterhalb des Elements `<right>` (`id<0`) bzw. `<left>` (`id>0`) erstellt. Als Grundlage für die nachfolgenden Konzept-Schritte wird dabei eine Datenstruktur erstellt, die für jeden BSSD-Fahrstreifen einen Eintrag enthält (`id` der übergeordneten Straße, `s`-Koordinate des übergeordneten BSSD-Segments, `id` des BSSD-Fahrstreifens). Diese Datenstruktur wird im Schaubild der Implementierung des fünften Konzept-Schritts (siehe Abbildung 6-9) als „BSSD-Fahrstreifen“ bezeichnet.

### 6.3.6 Schritt 6: Verknüpfen der OpenDRIVE- & BSSD-Fahrstreifen

Im sechsten Konzept-Schritt „Verknüpfen der OpenDRIVE- & BSSD-Fahrstreifen“ werden die zuvor erstellten BSSD-Fahrstreifen anhand einer Hilfsstruktur explizit mit den zugehörigen OpenDRIVE-Fahrstreifen verknüpft (siehe Abschnitt 5.3.1.6). Das Funktionsschaubild des sechsten Konzept-Schritts ist in Abbildung 6-10 dargestellt, die den zu diesem Schritt zugehörigen Teil des Gesamtschaubilds der Implementierung beinhaltet.

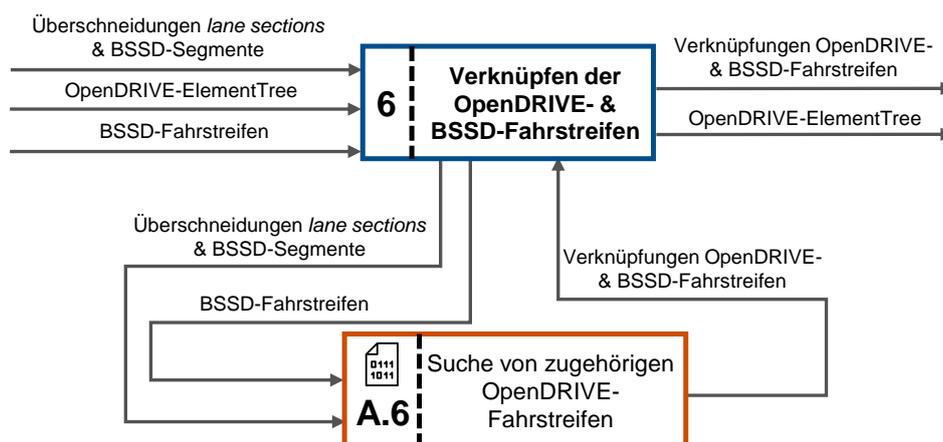


Abbildung 6-10: Funktionsschaubild Schritt 6 „Suche von zugehörigen OpenDRIVE-Fahrstreifen“

Die Implementierung des sechsten Konzept-Schritts besteht aus zwei aufeinanderfolgenden Teilen, die im Folgenden näher erläutert werden.

---

### 6.3.6.1 Algorithmus A.6: Suche von zugehörigen OpenDRIVE-Fahrstreifen

Der erste Teil des sechsten Konzept-Schritts ist die Durchführung des Algorithmus A.6 „Verknüpfen der OpenDRIVE- & BSSD-Fahrstreifen“. Bei diesem Algorithmus werden für jeden zuvor erstellten BSSD-Fahrstreifen alle OpenDRIVE-Fahrstreifen gesucht, die mit diesem BSSD-Fahrstreifen überschneiden. Dazu wird durch alle BSSD-Fahrstreifen iteriert (Datenstruktur „BSSD-Fahrstreifen“), wobei jeweils anhand der Datenstruktur „Überschneidungen *lane sections* & BSSD-Segmente“ alle *lane sections* herausgesucht werden, die zu dem übergeordneten BSSD-Segment überschneiden. Ausgehend von der ersten überschneidenden *lane section* wird in jeder überschneidenden *lane section* der zu dem betrachteten BSSD-Fahrstreifen äquivalente OpenDRIVE-Fahrstreifen herausgesucht. In der ersten überschneidenden *lane section* erfolgt dies über die äquivalente *id*. In den nachfolgenden *lane sections* ergeben sich die äquivalenten OpenDRIVE-Fahrstreifen über die Verknüpfung der OpenDRIVE-Fahrstreifen durch das Element `<successor>` (siehe beispielsweise Abbildung 5-10 und Tabelle 5-7). Für jeden OpenDRIVE-Fahrstreifen, der mit dem betrachteten BSSD-Fahrstreifen überschneidet, wird ein Eintrag in einer Datenstruktur erstellt. Diese Datenstruktur wird im Schaubild der Implementierung des sechsten Konzept-Schritts (siehe Abbildung 6-10) als „**Verknüpfungen OpenDRIVE- & BSSD-Fahrstreifen**“ bezeichnet.

### 6.3.6.2 Modifikation des OpenDRIVE-ElementTree

Im zweiten Teil des sechsten Konzept-Schritts wird im OpenDRIVE-*ElementTree* in jedem BSSD-`<lane>`-Element die explizite Verknüpfung zu den zu diesem BSSD-Fahrstreifen äquivalenten OpenDRIVE-Fahrstreifen hergestellt. Dazu wird innerhalb jedes BSSD-Fahrstreifens das Element `<assignLaneOpenDRIVE>` erstellt. Unterhalb dieses Elements wird für jeden OpenDRIVE-Fahrstreifen, der mit diesem BSSD-Fahrstreifen überschneidet (siehe Datenstruktur „Verknüpfungen OpenDRIVE- & BSSD-Fahrstreifen“), ein Element `<linkedLane>` erstellt. Dabei werden die Werte der Attribute `sLaneSection` und `id` entsprechend des überschneidenden OpenDRIVE-Fahrstreifens festgelegt. Anhand der erstellten Elemente ist an jeder Position der OpenDRIVE-Karte, bei der ein befahrbarer OpenDRIVE-Fahrstreifen definiert ist, eine eindeutige Zuordnung zu einem BSSD-Fahrstreifen und somit zu den Verhaltensattributen möglich.

## 6.3.7 Schritt 7: Erstellen der Verhaltensraum-Minimalstruktur

Im siebten Konzept-Schritt wird im OpenDRIVE-*ElementTree* unterhalb jedes BSSD-`<lane>`-Elements die Verhaltensraum-Minimalstruktur (siehe Abbildung A-2) eingefügt. Somit ist der OpenDRIVE-*ElementTree* in jeder Straße um eine Struktur erweitert, welche die Festlegung der Ausprägungen der einzelnen Verhaltensattribute in den BSSD-Segmenten dieser Straße ermöglicht.

## 6.3.8 Schritt 8: Befüllen des Verhaltensattributs Speed Limit

Im achten Konzept-Schritt „Befüllen des Verhaltensattributs *Speed Limit*“ werden nach Möglichkeit die Ausprägungen des Verhaltensattributs *Speed Limit* in den einzelnen BSSD-Fahrstreifen entlang und entgegen der Referenzrichtung extrahiert. Das Funktionsschaubild des achten Konzept-Schritts

ist in Abbildung 6-11 dargestellt, die den zu diesem Schritt zugehörigen Teil des Gesamtschaubilds der Implementierung beinhaltet.

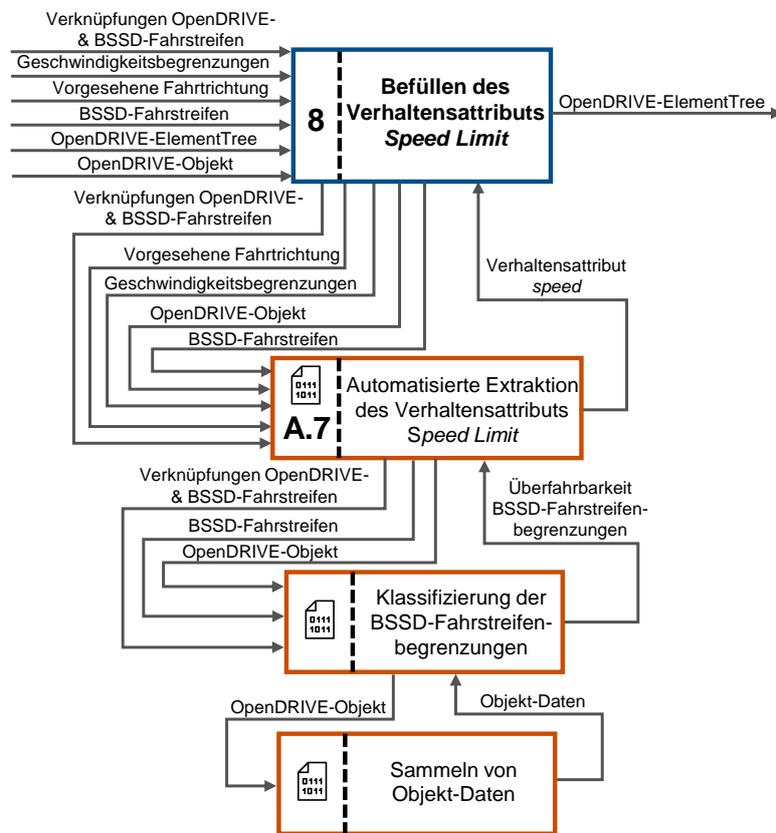


Abbildung 6-11: Funktionsschaubild Schritt 8 „Befüllen des Verhaltensattributs *Speed Limit*“

Die Implementierung des achten Konzept-Schritts besteht aus zwei aufeinanderfolgenden Teilen, die im Folgenden näher erläutert werden.

### 6.3.8.1 Algorithmus A.7: Automatisierte Extraktion des Verhaltensattributs *Speed Limit*

Der erste Teil des achten Konzept-Schritts ist die Durchführung des Algorithmus A.7 „Automatisierte Extraktion des Verhaltensattributs *Speed Limit*“. Bei diesem Algorithmus wird basierend auf den in der OpenDRIVE-Karte bezüglich der zulässigen Höchstgeschwindigkeit enthaltenen Informationen versucht, für jeden BSSD-Fahrstreifen die Ausprägung von *Speed Limit* entlang und entgegen der Referenzrichtung zu bestimmen. Die Ausführung des Algorithmus A.7 ist in zwei aufeinanderfolgende Schritte unterteilbar. Im ersten Teil wird *Speed Limit* für jeden BSSD-Fahrstreifen **entlang der vorgesehenen Fahrtrichtung** bestimmt, während im zweiten Teil die Bestimmung **entgegen der vorgesehenen Fahrtrichtung** erfolgt. Der Zusammenhang zwischen der vorgesehenen Fahrtrichtung und der Referenzrichtung wird anhand von Abbildung 6-12 erläutert, die zwei BSSD-Fahrstreifen mit entgegengesetzten Fahrtrichtungen beim Gelten von Rechtsverkehr zeigt. Die Referenzrichtung einer Straße in OpenDRIVE ist durch das Referenzlinien-Koordinatensystem der OpenDRIVE-Karte definiert. Die vorgesehene Fahrtrichtung in einem BSSD-Fahrstreifen ergibt sich durch das Gelten von Rechts- oder Linksverkehr. Ausgehend von der *id* eines BSSD-Fahrstreifens, die eine Zuordnung zu einer Straßenseite ermöglicht (rechts bei negativem Vorzeichen, links bei positivem

Vorzeichen), und der Kenntnis von Rechts- oder Linksverkehr ist somit für jeden BSSD-Fahrstreifen der Zusammenhang zwischen der vorgesehenen Fahrtrichtung und der Referenzrichtung bestimmbar. So ergibt sich bei Betrachtung des BSSD-Fahrstreifens mit der *id* -1, dass die vorgesehene Fahrtrichtung entlang der Referenzrichtung verläuft. Beim BSSD-Fahrstreifen mit der *id* 1 hingegen verläuft die vorgesehene Fahrtrichtung entgegen der Referenzrichtung. Bei Linksverkehr sind die beiden vorigen Aussagen andersherum gültig.

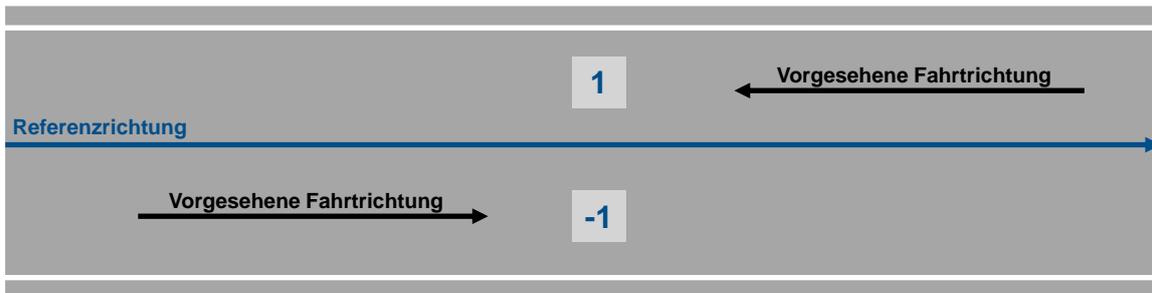


Abbildung 6-12: Zusammenhang zwischen vorgesehener Fahrtrichtung und Referenzrichtung

### 1. Extraktion *Speed Limit* entlang der vorgesehenen Fahrtrichtung

Die Bestimmung von *Speed Limit* entlang der vorgesehenen Fahrtrichtung erfolgt basierend auf der Datenstruktur „Geschwindigkeitsbegrenzungen“ (siehe Abschnitt 6.3.3.1), die Informationen über die einzelnen in den OpenDRIVE-Fahrstreifen geltenden zulässigen Höchstgeschwindigkeiten enthält. Es wird durch jeden BSSD-Fahrstreifen iteriert und dabei anhand der Datenstruktur „Geschwindigkeitsbegrenzungen“ für den ersten zu diesem BSSD-Fahrstreifen überschneidenden OpenDRIVE-Fahrstreifen überprüft, ob eine zulässige Höchstgeschwindigkeit für diesen OpenDRIVE-Fahrstreifen innerhalb des Definitionsbereichs des betrachteten BSSD-Fahrstreifens definiert ist. Es erfolgt nur die Betrachtung des ersten überschneidenden OpenDRIVE-Fahrstreifens, da die Segmentierung basierend auf den Geschwindigkeitsbegrenzungen durchgeführt wird (siehe Regel 6, Abschnitt 6.3.3.1). Somit ist sichergestellt, dass die zulässige Höchstgeschwindigkeit in allen zu dem BSSD-Fahrstreifen überschneidenden OpenDRIVE-Fahrstreifen identisch ist. Sofern der erste überschneidende OpenDRIVE-Fahrstreifen innerhalb des Definitionsbereichs des betrachteten BSSD-Fahrstreifens eine zulässige Höchstgeschwindigkeit besitzt, wird diese als Wert von *Speed Limit* entlang der vorgesehenen Fahrtrichtung für den aktuellen BSSD-Fahrstreifen gespeichert. Je nach Rechts- oder Linksverkehr und der Straßenseite des BSSD-Fahrstreifens entspricht dies der Ausprägung von *Speed Limit* entlang oder entgegen der Referenzrichtung (siehe Abbildung 6-12). Zur Speicherung der Ausprägungen von *Speed Limit* wird eine Datenstruktur erstellt, die für jeden BSSD-Fahrstreifen diese Ausprägung entlang und entgegen der Referenzrichtung speichert. Diese Datenstruktur wird im Schaubild der Implementierung des achten Konzept-Schritts (siehe Abbildung 6-11) als „**Verhaltensattribut *Speed Limit***“ bezeichnet. Sofern es nicht möglich ist, für einen BSSD-Fahrstreifen den Wert von *Speed Limit* zu bestimmen, wird in der Datenstruktur der Wert "None" hinterlegt.

### 2. Extraktion *Speed Limit* entgegen der vorgesehenen Fahrtrichtung

Der Verhaltensraum und somit auch das Verhaltensattribut *Speed Limit* wird entgegen der vorgesehenen Fahrtrichtung festgelegt, um eine Definition der BSSD in jeder Situation zu ermöglichen. So

ist beispielweise bei einem in den Gegenverkehr geratenen Fahrzeug erforderlich, dass dieses nach wie vor Kenntnis über die Ausprägungen der Verhaltensattribute besitzt. Daraus ergibt sich, dass für die Bestimmung von *Speed Limit* entgegen der vorgesehenen Fahrtrichtung verschiedene Fälle zu unterscheiden sind, die sich darauf beziehen, ob es möglich ist, dass ein Fahrzeug in den Gegenverkehr gerät oder nicht.

Zur Bestimmung von *Speed Limit* entgegen der vorgesehenen Fahrtrichtung wird durch jeden BSSD-Fahrstreifen iteriert und eine Überprüfung verschiedener Fälle durchgeführt, die im Folgenden erläutert werden. Zunächst wird überprüft, ob sich der betrachtete BSSD-Fahrstreifen innerhalb einer Einbahnstraße befindet. Dies ist der Fall, wenn alle BSSD-Fahrstreifen des übergeordneten BSSD-Segments das gleiche Vorzeichen des Attributs *id* besitzen. Bei einer Einbahnstraße existiert kein Gegenverkehr und daher entspricht die Ausprägung von *Speed Limit* entgegen der vorgesehenen Fahrtrichtung der Ausprägung entlang der vorgesehenen Fahrtrichtung. Der Wert für *Speed Limit* in der Datenstruktur „Verhaltensattribut *Speed Limit*“ wird dementsprechend gesetzt. Falls der betrachtete BSSD-Fahrstreifen in keiner Einbahnstraße definiert ist, wird überprüft, ob es möglich ist, dass ein Fahrzeug, das innerhalb dieses BSSD-Fahrstreifens fährt, in den Gegenverkehr gerät oder nicht. Falls dies für einen BSSD-Fahrstreifen möglich bzw. nicht möglich ist, wird dieser im Folgenden als *nicht vom Gegenverkehr getrennt* bzw. *vom Gegenverkehr getrennt* bezeichnet. Wenn der betrachtete BSSD-Fahrstreifen vom Gegenverkehr getrennt ist, wird die Ausprägung von *Speed Limit* analog zum Fall der Einbahnstraße festgelegt. Für den Fall, dass der betrachtete BSSD-Fahrstreifen nicht vom Gegenverkehr getrennt ist, entspricht die Ausprägung von *Speed Limit* entgegen der vorgesehenen Fahrtrichtung der Ausprägung des innersten BSSD-Fahrstreifens der Gegenrichtung entlang der vorgesehenen Fahrtrichtung. Zur Verdeutlichung der Festlegung von *Speed Limit* entlang und entgegen der Referenzrichtung zeigt Abbildung 6-13 zwei BSSD-Fahrstreifen mit unterschiedlichen Ausprägungen von *Speed Limit* entlang der vorgesehenen Fahrtrichtung. Die aus dieser Szenerie resultierenden Ausprägungen von *Speed Limit* für die jeweiligen BSSD-Fahrstreifen für den Fall, dass die BSSD-Fahrstreifen vom Gegenverkehr getrennt bzw. nicht getrennt sind, sind in Tabelle 6-1 gelistet.



Abbildung 6-13: Festlegung der Ausprägung des Verhaltensattributs *Speed Limit*

Tabelle 6-1: Ausprägungen Verhaltensattribut *Speed Limit* für Szenerie aus Abbildung 6-13

id	Vom Gegenverkehr getrennt		Nicht vom Gegenverkehr getrennt	
	<i>Speed Limit</i> behavior-Along	<i>Speed Limit</i> behavior-Against	<i>Speed Limit</i> behavior-Along	<i>Speed Limit</i> behavior-Against
-1	30	30	30	50
1	50	50	30	50

Die Überprüfung, ob ein BSSD-Fahrstreifen vom Gegenverkehr getrennt ist, erfolgt anhand von zwei Kriterien. Sofern eines dieser beiden Kriterien erfüllt ist, wird der BSSD-Fahrstreifen als vom Gegenverkehr getrennt klassifiziert. Das erste Kriterium

Kriterium 1: Ein BSSD-Fahrstreifen ist vom Gegenverkehr getrennt, wenn sich ein nicht-befahrbarer OpenDRIVE-Fahrstreifen zwischen den beiden innersten BSSD-Fahrstreifen des übergeordneten BSSD-Segments befindet

trifft die Annahme, dass ein BSSD-Fahrstreifen vom Gegenverkehr getrennt ist, wenn sich in der Mitte der Straße ein für Fahrzeuge nicht befahrbarer Bereich befindet. Eine beispielhafte Szenerie dazu ist in Abbildung 6-14 gezeigt. Zwischen den beiden BSSD-Fahrstreifen (id 1 und -2) befindet sich ein nicht-befahrbarer OpenDRIVE-Fahrstreifen, der beispielsweise eine Verkehrsinsel repräsentiert. Diese Verkehrsinsel führt dazu, dass die beiden BSSD-Fahrstreifen jeweils vom Gegenverkehr getrennt sind.

Das zweite Kriterium für die Feststellung der Trennung eines BSSD-Fahrstreifens vom Gegenverkehr

Kriterium 2: Ein BSSD-Fahrstreifen ist vom Gegenverkehr getrennt, wenn eine aus Sicht dieses BSSD-Fahrstreifens weiter innen liegende nicht-überfahrbare Begrenzung eines BSSD-Fahrstreifens existiert, die sich auf der gleichen Straßenseite wie der BSSD-Fahrstreifen befindet

basiert auf der Klassifizierung der lateralen Begrenzungen eines BSSD-Fahrstreifens in *überfahrbar* bzw. *nicht-überfahrbar*. Dabei ist eine BSSD-Fahrstreifenbegrenzung überfahrbar, wenn denkbar ist, dass ein Fahrzeug über diese Begrenzung auf den benachbarten Fahrstreifen<sup>83</sup> gelangt. Eine beispielhafte Szenerie, bei der Kriterium 2 Anwendung findet, ist in Abbildung 6-15 dargestellt. Zwischen den BSSD-Fahrstreifen mit der id -1 und -2 befindet sich eine Leitplanke, wie sie beispielsweise in Baustellen zum Einsatz kommt. Diese Leitplanke führt dazu, dass die BSSD-Fahrstreifenbegrenzung zwischen diesen beiden BSSD-Fahrstreifen nicht-überfahrbar ist. Entsprechend Kriterium 2 ist der BSSD-Fahrstreifen mit der id -2 vom Gegenverkehr getrennt, wohingegen der BSSD-Fahrstreifen mit der id -1 nicht vom Gegenverkehr getrennt ist.

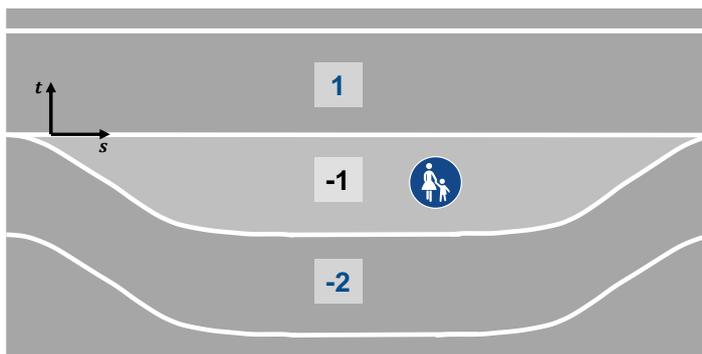


Abbildung 6-14: Trennung eines BSSD-Fahrstreifens vom Gegenverkehr, Kriterium 1

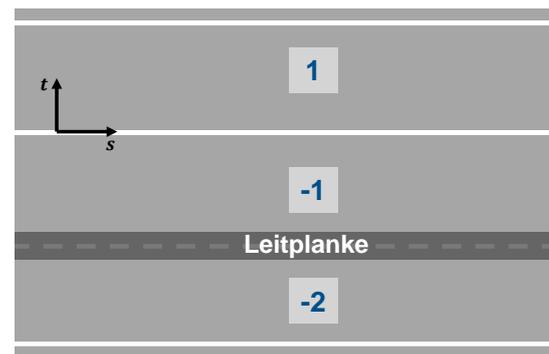


Abbildung 6-15: Trennung eines BSSD-Fahrstreifens vom Gegenverkehr, Kriterium 2

Die für Kriterium 2 benötigte Klassifizierung der Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen wird in einem separaten Algorithmus durchgeführt, der im Folgenden näher erläutert wird.

<sup>83</sup> In diesem Fall werden sowohl benachbarte BSSD-Fahrstreifen als auch benachbarte OpenDRIVE-Fahrstreifen betrachtet. Aufgrund dessen wird in diesem Kontext nur der Begriff *Fahrstreifen* verwendet.

## Algorithmus: Klassifizierung der BSSD-Fahrstreifenbegrenzungen

Der Algorithmus „Klassifizierung der BSSD-Fahrstreifenbegrenzungen“ klassifiziert die beiden lateralen Begrenzungen eines BSSD-Fahrstreifens (linke und rechte Begrenzung) als überfahrbar bzw. nicht-überfahrbar. Dies erfolgt in zwei aufeinanderfolgenden Schritten.

Im ersten Schritt erfolgt die Klassifizierung der Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen anhand der zu dem BSSD-Fahrstreifen benachbarten Fahrstreifen. Dazu wird durch jeden BSSD-Fahrstreifen iteriert und die Art (Attribut `type`) des zu diesem BSSD-Fahrstreifen linken und rechten benachbarten Fahrstreifens betrachtet. Ist der benachbarte Fahrstreifen existent und wird durch ein `type`-Attribut repräsentiert, das nicht die Ausprägungen "none" (Keine reale Repräsentation) oder "curb" (Bordstein) besitzt, wird die BSSD-Fahrstreifenbegrenzung zu diesem benachbarten Fahrstreifen als überfahrbar klassifiziert. Folglich erfolgt bei den Ausprägungen "none" und "curb" des benachbarten Fahrstreifens eine Klassifizierung als nicht-überfahrbare BSSD-Fahrstreifenbegrenzung. Wenn kein benachbarter Fahrstreifen existiert, was bei den äußersten Fahrstreifen einer Straße auftritt, wird die zugehörige BSSD-Fahrstreifenbegrenzung ebenfalls auf nicht-überfahrbar gesetzt. Bei der Durchführung dieses ersten Schritts wird eine Datenstruktur erstellt, die für jeden BSSD-Fahrstreifen die Klassifizierung der Überfahrbarkeit der linken und rechten Begrenzung dieses BSSD-Fahrstreifens enthält. Diese Datenstruktur wird im Schaubild der Implementierung des achten Konzept-Schritts (siehe Abbildung 6-11) als „Überfahrbarkeit BSSD-Fahrstreifenbegrenzungen“ bezeichnet. Die Anwendung der Klassifizierung der Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen anhand des benachbarten Fahrstreifens wird anhand von Abbildung 6-16 gezeigt. Die in der Abbildung gezeigte Szenerie beinhaltet zwei BSSD-Fahrstreifen sowie die Klassifizierung der Überfahrbarkeit der zugehörigen Begrenzungen. Überfähre Begrenzungen sind in der Abbildung grün markiert, während nicht-überfähre Begrenzungen rot markiert sind. Die linke Begrenzung des BSSD-Fahrstreifens mit der `id` 1 ist nicht-überfahrbar, da kein linker benachbarter Fahrstreifen existiert. Die rechte Begrenzung (und somit auch die linke Begrenzung des BSSD-Fahrstreifens mit der `id` -1) ist aufgrund eines benachbarten Fahrstreifens vom `type`="driving" überfahrbar. Die rechte Begrenzung des BSSD-Fahrstreifens mit der `id` -1 ist aufgrund eines rechten benachbarten Fahrstreifens vom `type`="curb" nicht-überfahrbar.



Abbildung 6-16: Klassifizierung Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen anhand der Nachbarfahrstreifen

Im zweiten Schritt wird die im ersten Schritt getätigte Klassifizierung der Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen anhand der in der OpenDRIVE-Karte vorhandenen Objekte modifiziert. Als Grundlage dafür wird der Algorithmus „Sammeln von Objekt-Daten“ durchgeführt, der im Folgenden näher erläutert wird

### Algorithmus: Sammeln von Objekt-Daten

Der Algorithmus „Sammeln von Objekt-Daten“ extrahiert Informationen über alle Objekte in der OpenDRIVE-Karte, die relevant für die Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen sind. Dabei beinhaltet der Begriff *Objekte* sowohl Objekte, die durch das Element <object> repräsentiert werden als auch Objekte, die durch das Element <repeat> repräsentiert werden (siehe Abschnitt 3.3.1.6). Für die Durchführung dieses Algorithmus wird durch jedes <object>- sowie untergeordnete <repeat>-Element iteriert. Dabei wird zunächst anhand von zwei Bedingungen überprüft, ob das betrachtete Objekt für die Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen relevant ist. Einerseits ist erforderlich, dass sich der Ursprung des Objekts auf einem OpenDRIVE-Fahrstreifen befindet<sup>84</sup>. Für die Überprüfung dieser Bedingung wird ein separater Algorithmus ausgeführt, der überprüft, ob es möglich ist, den Ursprung des Objekts (Attribute *s*, *t*) einem OpenDRIVE-Fahrstreifen zuzuordnen und ggf. diesen OpenDRIVE-Fahrstreifen zurückgibt. Andererseits ist das Objekt für die Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen relevant, wenn es sich in einer Höhe befindet, die für ein Fahrzeug relevant ist. Diese Höhe wird entsprechend §22 Absatz 2 StVO<sup>85</sup> festgelegt, der eine maximale Höhe von Fahrzeug und Ladung von 4 m vorschreibt. Diese maximale Höhe ist veränderbar und somit an das aus BSSD-Sicht betrachtete Ego-Fahrzeug anpassbar.

Sofern die beiden genannten Bedingungen erfüllt sind, werden zu dem betrachteten Objekt verschiedene Daten erhoben, die in einer Datenstruktur gespeichert werden. Diese Datenstruktur wird im Schaubild der Implementierung des achten Konzept-Schritts (siehe Abbildung 6-11) als „**Objekt-Daten**“ bezeichnet. Die darin enthaltenen Daten sind in Tabelle 6-2 gelistet und erklärt.

Tabelle 6-2: Datenstruktur „Objekt-Daten“

Daten	Beschreibung
object_id	id des <object>-Elements bzw. des übergeordneten <object>-Elements (Bei Element <repeat>)
road_id	id der Straße innerhalb der Objektursprung definiert ist
laneSection_s	s-Koordinate der <i>lane section</i> innerhalb der Objektursprung definiert ist
lane_id	id des OpenDRIVE-Fahrstreifens innerhalb dessen Objektursprung definiert ist
s_origin	s-Koordinate des Objektursprungs
t_origin	t-Koordinate des Objektursprungs
delta_t_left	Abstand des Objektursprungs in <i>t</i> -Richtung zu linker Begrenzung des zugehörigen OpenDRIVE-Fahrstreifens
delta_t_right	Abstand des Objektursprungs in <i>t</i> -Richtung zu rechter Begrenzung des zugehörigen OpenDRIVE-Fahrstreifens
s_min	Minimale s-Koordinate der Ausdehnung des Objekts
s_max	Maximale s-Koordinate der Ausdehnung des Objekts

<sup>84</sup> Oftmals werden Objekte außerhalb der lateralen Grenzen einer Straße platziert, wie beispielsweise Leitplanken auf einer Autobahn.

<sup>85</sup> BMJV: Straßenverkehrs-Ordnung (StVO) (2013), §22 Absatz 2.

Die Daten `object_id`, `road_id`, `s_origin` und `t_origin` sind direkt aus den Attributen des betrachteten `<object>`- bzw. `<repeat>`-Elements extrahierbar (siehe Abschnitt 3.3.1.6). Die Daten `laneSection_s` und `lane_id` ergeben sich aus dem angesprochenen Algorithmus, der den Ursprung eines Objekts einem OpenDRIVE-Fahrstreifen zuordnet. Als zusätzliche Funktionalität berechnet dieser Algorithmus den lateralen Abstand des Ursprungs des Objekts zu der linken und rechten Begrenzung des zugeordneten OpenDRIVE-Fahrstreifens, woraus sich die Daten `delta_t_left` und `delta_t_right` ergeben. Für die Berechnung der Daten `s_min` und `s_max` ist, im Gegensatz zu den bisherigen Daten, die Betrachtung der Abmessung und der Orientierung des Objekts notwendig. Dazu wird zunächst anhand der dem Objekt zugehörigen Attribute (`length`, `width`, `height`, `radius`) die Ausdehnung des Objekts in  $u$ ,  $v$ - und  $z$ -Richtung bestimmt. Daraus ergeben sich acht Eckpunkte, welche die Objektausdehnung in Form einer Bounding Box beschreiben. Diese acht Eckpunkte werden anschließend anhand des Gierwinkels  $\gamma$ , des Nickwinkels  $\beta$  und des Rollwinkels  $\alpha$ , welche die Ausrichtung des Objekts definieren, vom Lokal-Koordinatensystem in das Referenzlinien-Koordinatensystem transformiert. Durch Betrachtung der Drehfolge Gieren, Nicken, Rollen ergibt sich die Transformationsbeziehung

$$\begin{pmatrix} s \\ t \\ h \end{pmatrix} = \begin{pmatrix} \cos \beta \cdot \cos \gamma & -\cos \alpha \cdot \sin \gamma + \cos \gamma \cdot \sin \alpha \cdot \sin \beta & \sin \alpha \cdot \sin \gamma + \cos \alpha \cdot \cos \gamma \cdot \sin \beta \\ \cos \beta \cdot \sin \gamma & \cos \alpha \cdot \cos \gamma + \sin \alpha \cdot \sin \beta \cdot \sin \gamma & -\cos \gamma \cdot \sin \alpha + \cos \alpha \cdot \sin \beta \cdot \sin \gamma \\ -\sin \beta & \cos \beta \cdot \sin \alpha & \cos \alpha \cdot \cos \beta \end{pmatrix} \cdot \begin{pmatrix} u \\ v \\ z \end{pmatrix} \quad (6-1)$$

zur Umrechnung der Eckpunkte in das Referenzlinien-Koordinatensystem. Basierend darauf ergibt sich die Ausdehnung des betrachteten Objekts in  $s$ -Richtung. Durch Subtraktion bzw. Addition zu der  $s$ -Koordinate des Ursprungs des Objekts werden `s_min` und `s_max` berechnet. Zur Verdeutlichung der über ein Objekt gesammelten Daten zeigt Abbildung 6-17 ein beispielhaftes Objekt, das sich innerhalb eines OpenDRIVE-Fahrstreifens befindet, sowie Tabelle 6-3 die daraus resultierenden Werte der Datenstruktur „Objekt-Daten“.

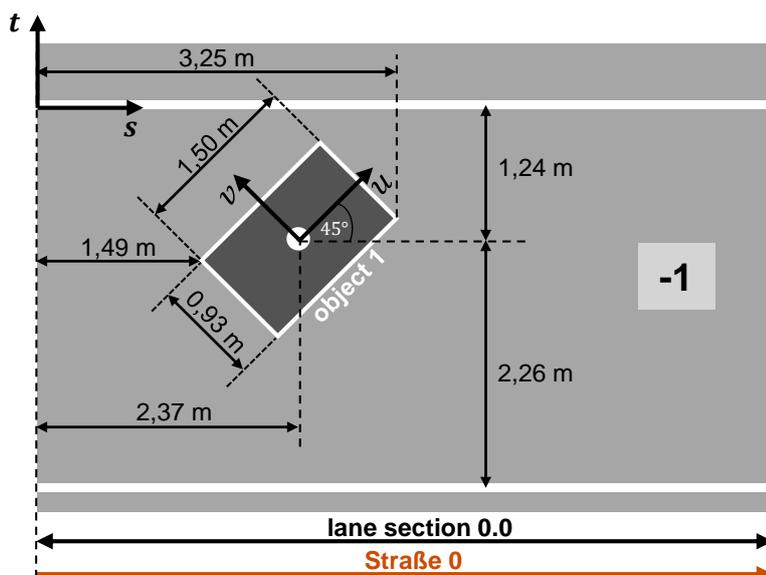


Tabelle 6-3: Werte „Objekt-Daten“ für Objekt aus Abbildung 6-17

Daten	Wert
<code>object_id</code>	1
<code>road_id</code>	0
<code>laneSection_s</code>	0.0
<code>lane_id</code>	-1
<code>s_origin</code>	2.37
<code>t_origin</code>	-1.24
<code>delta_t_left</code>	1.24
<code>delta_t_right</code>	2.26
<code>s_min</code>	1.49
<code>s_max</code>	3.25

Abbildung 6-17: Objekt innerhalb eines OpenDRIVE-Fahrstreifens

Nachdem die Daten aller Objekte, die für eine Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen relevant sind, berechnet und in die Datenstruktur „Objekt-Daten“ eingetragen wurden, wird diese

---

Datenstruktur der Funktion zur Klassifizierung der Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen zurückgegeben. Diese Funktion iteriert durch alle BSSD-Fahrstreifen und greift dabei anhand der Datenstruktur „Objekt-Daten“ auf alle Objekte zu, die in longitudinaler und lateraler Richtung innerhalb des betrachteten BSSD-Fahrstreifens definiert sind. In longitudinaler Richtung sind dies alle Objekte, deren Ausdehnung in *s*-Richtung sich innerhalb des dem BSSD-Fahrstreifen übergeordneten BSSD-Segments befinden. In lateraler Richtung werden alle Objekte betrachtet, deren Ursprung sich innerhalb eines OpenDRIVE-Fahrstreifens befindet, der den betrachteten BSSD-Fahrstreifen repräsentiert. Dies umfasst sowohl OpenDRIVE-Fahrstreifen, die direkt mit dem BSSD-Fahrstreifen verknüpft sind (Element `<linkedLane>`) als auch OpenDRIVE-Fahrstreifen, die indirekt über die Angabe von Vorgänger- bzw. Nachfolger-Fahrstreifen (Elemente `<predecessor>/<successor>`) mit einem direkt mit dem BSSD-Fahrstreifen verknüpften OpenDRIVE-Fahrstreifen verknüpft sind.

Anhand des Abstands der Objekte zu der linken bzw. rechten Begrenzung des zugehörigen OpenDRIVE-Fahrstreifens (`delta_t_left`, `delta_t_right`) erfolgt eine Zuordnung der Objekte zu der linken und rechten Begrenzung des betrachteten BSSD-Fahrstreifens. Dabei wird ein Objekt einer Begrenzung zugeordnet, wenn der laterale Abstand des Ursprungs dieses Objekts zu dieser Begrenzung kleiner als 0,5 m ist. Dieser Schwellwert verhindert, dass Objekte, die sich in der Mitte eines BSSD-Fahrstreifens befinden, einer bestimmten Begrenzung zugeordnet werden, erlaubt aber gleichzeitig einen gewissen Versatz von Objekten, die das Überfahren einer BSSD-Fahrstreifenbegrenzung verhindern. Nachdem alle einem BSSD-Fahrstreifen zugehörigen Objekte der linken, rechten oder gar keiner Begrenzung dieses BSSD-Fahrstreifens zugeordnet sind, wird überprüft, ob die Objekte zu einer nicht-überfahrbaren Begrenzung führen. Dazu wird für die linke und rechte Begrenzung jeweils geprüft, ob über den gesamten Definitionsbereich der *s*-Koordinate des betrachteten BSSD-Fahrstreifens mindestens ein der linken bzw. rechten Begrenzung zugeordnetes Objekt definiert ist. Sofern dies der Fall ist, wird die entsprechende BSSD-Fahrstreifenbegrenzung als nicht-überfahrbar in der Datenstruktur „Überfahrbarkeit BSSD-Fahrstreifenbegrenzungen“ gesetzt.

Zur Veranschaulichung der Klassifizierung der Überfahrbarkeit von BSSD-Fahrstreifenbegrenzungen basierend auf Objekten zeigt Abbildung 6-18 eine Szenerie mit einem BSSD-Segment, bestehend aus drei BSSD-Fahrstreifen und mehreren Objekten. Die Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen ist anhand der grünen (überfahrbar) und roten (nicht-überfahrbar) Strichlinien gekennzeichnet. Ein Objekt der Szenerie repräsentiert eine Leitplanke, welche die BSSD-Fahrstreifen mit der `id -1` und `-2` über den gesamten Definitionsbereich des BSSD-Segments trennt. Folglich ist die BSSD-Fahrstreifenbegrenzung zwischen diesen beiden BSSD-Fahrstreifen nicht-überfahrbar. In der Mitte der Straße befinden sich mehrere Laternen. Diese sind jedoch nicht kontinuierlich über den gesamten Definitionsbereich des BSSD-Segments definiert. Aufgrund dessen ist die rechte bzw. linke Begrenzungen des BSSD-Fahrstreifens mit der `id 2` bzw. `-1` überfahrbar.

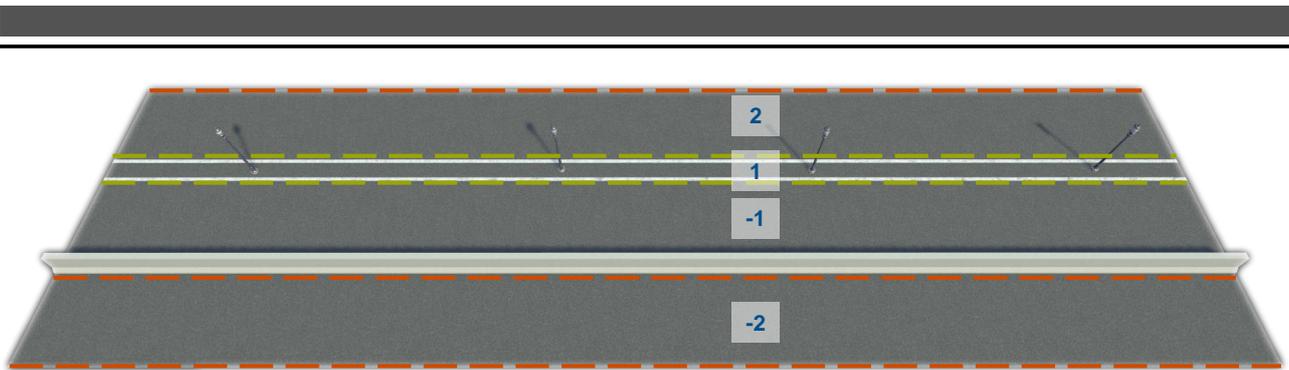


Abbildung 6-18: Klassifizierung Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen anhand von Objekten

### 6.3.8.2 Modifikation des OpenDRIVE-ElementTree

Im zweiten Teil des achten Konzept-Schritts werden die zuvor im Algorithmus A.7 extrahierten Werte des Verhaltensattributs *Speed Limit* in die OpenDRIVE-Datei eingetragen. Dazu wird durch jedes `<lane>`-Element iteriert und die in der Datenstruktur „Verhaltensattribut *Speed Limit*“ hinterlegten Werte für *Speed Limit* entlang und entgegen der Referenzrichtung in das Attribut `max` der beiden zu den Elementen `<behaviorAlong>` und `<behaviorAgainst>` zugehörigen Elemente `<speed>` eingefügt.

### 6.3.9 Schritt 9: Abspeichern der modifizierten OpenDRIVE-Datei

Im neunten Konzept-Schritt „Abspeichern der modifizierten OpenDRIVE-Datei“ wird der durch die vorangegangenen Konzept-Schritte modifizierte OpenDRIVE-*ElementTree* als XODR-Datei abgespeichert. Somit existiert eine OpenDRIVE-Datei, die um die im Konzept definierte BSSD-Struktur erweitert ist, und die Ausführung des Software-Tools ist abgeschlossen. Die Spezifikation der in dieser Arbeit umgesetzten BSSD-Integration ist im Anhang A.5 zusammengefasst.

## 6.4 Testen der Implementierung

Um die korrekte Funktionalität der Implementierung möglichst vollständig sicherzustellen, werden Software-Tests definiert und implementiert. Dabei wird für jede Komponente der Implementierung ein separater Testfall definiert. Dies umfasst die Konzept-Schritte 1–9, die übergeordneten Algorithmen A.1–A.7 sowie einzelne in den zuvor genannten Funktionen verwendete Subfunktionen. Dabei beinhaltet ein Testfall immer mindestens einen Test. Das allgemeine Vorgehen zur Durchführung eines Tests wird im Folgenden beschrieben.

### 6.4.1 Vorgehen

Ein Test dient dem Überprüfen der Funktionalität einer bestimmten Komponente der Implementierung im Hinblick auf einen bestimmten Aspekt. Dies erfolgt nach einer einheitlichen Vorgehensweise in drei Schritten, die in Abbildung 6-19 dargestellt ist. Im ersten Schritt werden die zur Ausführung der zu testenden Komponente benötigten Eingabedaten bereitgestellt. Diese werden mit dem Ziel erzeugt, im Hinblick auf den zu testenden Aspekt möglichst herausfordernd zu sein. Im zweiten Schritt wird die Komponente mit den im ersten Schritt definierten Eingabedaten ausgeführt. Das

daraus resultierende Ergebnis wird im dritten Schritt mit dem erwarteten Ergebnis verglichen, das als Referenz für die korrekte Funktionsweise dient. Sofern das Ergebnis mit dem erwarteten Ergebnis übereinstimmt, ist der Test bestanden. Im anderen Fall gilt der Test als nicht bestanden.

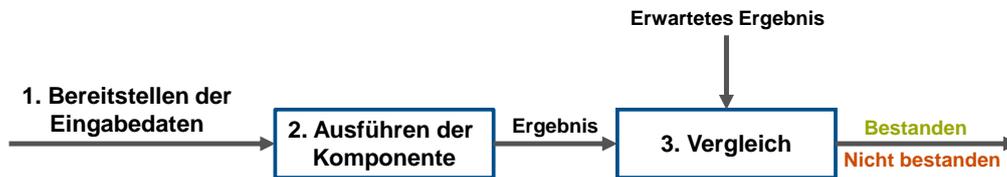


Abbildung 6-19: Vorgehen zur Durchführung eines Tests

Bei allen Tests, deren zu testende Komponente eine OpenDRIVE-Datei als Eingabe benötigt (OpenDRIVE-Objekt oder OpenDRIVE-*ElementTree*), wird die entsprechende Szenerie mit der Software *RoadRunner*<sup>86</sup> der Firma *MathWorks* erzeugt. *RoadRunner* bietet die Möglichkeit der manuellen Erstellung von Verkehrsnetzwerken und dem Export dieser in Form einer OpenDRIVE-Datei.

## 6.4.2 Betrachtung exemplarischer Tests

Eine vollständige Beschreibung aller im Rahmen dieser Arbeit definierten Testfälle und Tests überschreitet den Rahmen dieser Ausarbeitung. Aufgrund dessen werden im Folgenden zwei Tests beschrieben, um die Vorgehensweise exemplarisch zu demonstrieren.

### 6.4.2.1 Test 1: Klassifizierung der Fahrbahnzugehörigkeit von Fahrradwegen

Dieser Test wird im Rahmen des Testfalls zur Überprüfung des Algorithmus A.1 („Automatisierte Suche von befahrbaren Fahrstreifen“, siehe Abschnitt 6.3.2.1) durchgeführt. Es wird geprüft, ob OpenDRIVE-Fahrstreifen mit dem Attribut `type="biking"` (Fahrradweg) korrekt als befahrbar bzw. nicht-befahrbar basierend auf dem benachbarten OpenDRIVE-Fahrstreifen klassifiziert werden. Der zu testende Algorithmus A.1 benötigt als Eingabedaten ein OpenDRIVE-Objekt (siehe Abbildung 6-4). Folglich wird in *RoadRunner* eine Szenerie erstellt, welche die Klassifizierung der Befahrbarkeit von Fahrradwegen möglichst herausfordernd testet. Bei dieser Klassifizierung sind in Abhängigkeit der benachbarten OpenDRIVE-Fahrstreifen zwei Fälle zu unterscheiden. Wenn mindestens ein benachbarter OpenDRIVE-Fahrstreifen befahrbar ist, ist der Fahrradweg ebenfalls befahrbar. Andernfalls ist der Fahrradweg nicht-befahrbar. Daraus ergibt sich die in Abbildung 6-20 dargestellte Szenerie, welche diese beiden Fälle abbildet und als Eingang für diesen Test dient. Der OpenDRIVE-Fahrstreifen mit der `id -2` repräsentiert einen Fahrradschutzstreifen, der ein Teil der Fahrbahn darstellt und somit als befahrbar zu klassifizieren ist. Der OpenDRIVE-Fahrstreifen mit der `id 4` hingegen repräsentiert einen von der Straße separierten Fahrradweg, der aufgrund dessen als nicht-befahrbar zu klassifizieren ist. Das Resultat des Algorithmus A.1 ist die Datenstruktur „Befahrbarkeit OpenDRIVE-Fahrstreifen“. Somit ergibt sich das erwartete Ergebnis des Tests durch das manuelle Erzeugen dieser Datenstruktur aus der betrachteten Szenerie. Das erwartete Ergebnis des Tests ist in Tabelle 6-4 gelistet.

<sup>86</sup> <https://de.mathworks.com/products/roadrunner.html>

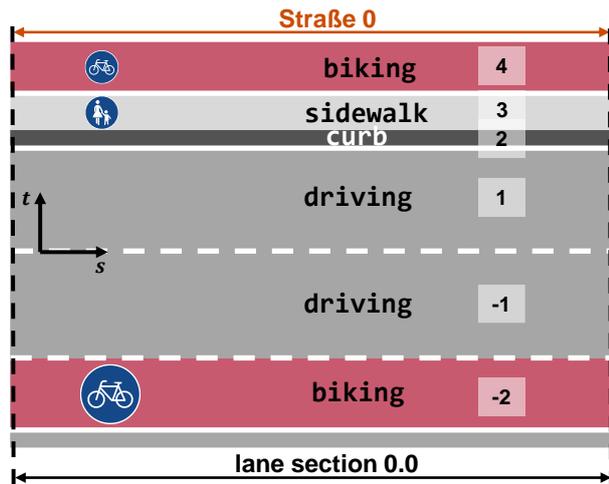


Tabelle 6-4: Erwartetes Ergebnis Test 1

road_id	lane-section	lane_id	type	drivable
0	0.0	-2	biking	True
0	0.0	-1	driving	True
0	0.0	1	driving	True
0	0.0	2	curb	False
0	0.0	3	sidewalk	False
0	0.0	4	biking	False

Abbildung 6-20: Szenerie Test 1

### 6.4.2.2 Test 2: Segmentierung basierend auf Verkehrszeichen

Dieser Test wird beim Testfall zur Überprüfung des Algorithmus A.3 („Automatisierte Extraktion von Segmenten“, siehe Abschnitt 6.3.3.1) durchgeführt. Es wird geprüft, ob BSSD-Segmente basierend auf den in einer OpenDRIVE-Karte enthaltenen Verkehrszeichen korrekt entsprechend Regel 7 für die Definition neuer BSSD-Segmente (Definition eines BSSD-Segments bei jedem in Deutschland gültigen Verkehrszeichen, das ein Indikationselement repräsentiert) gefunden werden. Die zu testende Funktion benötigt als Eingabedaten ein OpenDRIVE-Objekt. Aufgrund dessen wird in *RoadRunner* eine Szenerie erstellt, welche die Segmentierung basierend auf Verkehrszeichen möglichst herausfordernd testet. In diesem Kontext bedeutet herausfordernd, dass alle möglichen Fälle von Verkehrszeichen, welche die Definition eines BSSD-Segments erfordern bzw. nicht erfordern, abgeprüft werden. Daraus ergibt sich die in Abbildung 6-21 dargestellte Szenerie, die sowohl die Eingangsdaten als auch das erwartete Ergebnis zeigt. Alle BSSD-Segmente, die bereits in vorangegangenen Funktionen zur Extraktion von BSSD-Segmenten gefunden wurden, sind blau markiert. Alle durch die Funktion „Segmente basierend auf Verkehrszeichen“ gefundenen BSSD-Segmente sind gelb markiert.

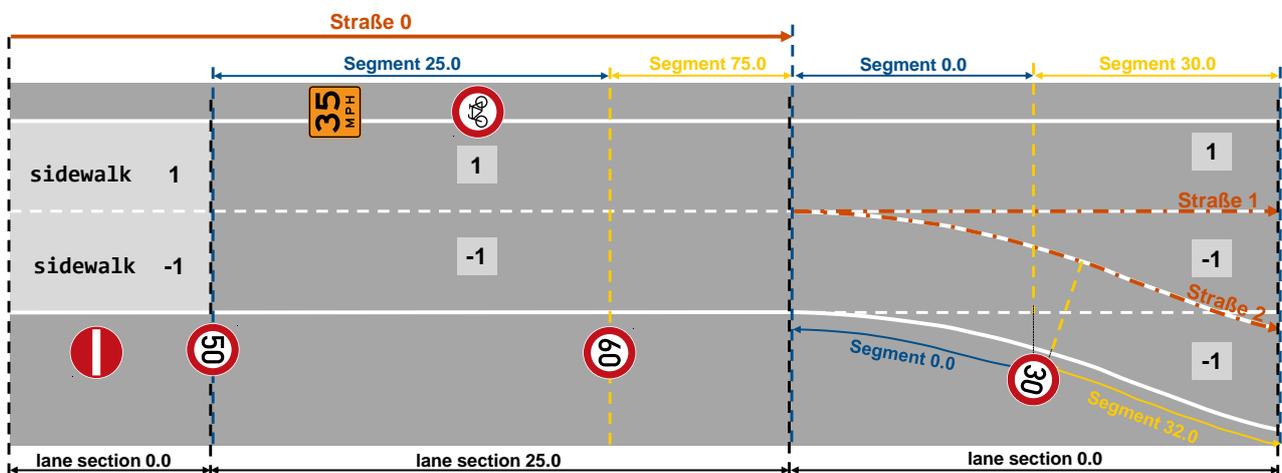


Abbildung 6-21: Szenerie und erwartetes Ergebnis Test 2

---

Die Szenerie besteht aus drei miteinander verbundenen Straßen. Die erste *lane section* von Straße 0 besteht aus zwei Gehwegen (`type="sidewalk"`). Folglich repräsentiert dieser Bereich eine BSSD-Definitionslücke. In diesem Bereich ist das Verkehrszeichen 267 („Verbot der Einfahrt“) definiert, welches ein Indikationselement repräsentiert. Da jedoch kein befahrbarer OpenDRIVE-Fahstreifen bei diesem Verkehrszeichen existiert, wird kein BSSD-Segment definiert. Die *lane section* 25.0 von Straße 0 besteht aus zwei befahrbaren OpenDRIVE-Fahstreifen (`type="driving"`). Aufgrund dessen wurde bereits durch die Funktion „Segmente basierend auf der Anzahl der befahrbaren Fahstreifen“ das Segment 25.0 definiert. Bei dieser *s*-Koordinate existiert zugleich ein Verkehrszeichen zur Festlegung einer zulässigen Höchstgeschwindigkeit von 50 km/h. Da an dieser Stelle jedoch bereits ein BSSD-Segment existiert, wird kein neues BSSD-Segment definiert. Im weiteren Verlauf der *lane section* ist ein in den USA verwendetes Verkehrszeichen zur Definition einer zulässigen Höchstgeschwindigkeit definiert. Dieses repräsentiert kein in Deutschland gültiges Verkehrszeichen und erfordert somit nicht die Definition eines neuen BSSD-Segments. Weiterhin wird das Verkehrszeichen 254 („Verbot für Radverkehr“) definiert, welches kein Indikationselement repräsentiert. Daher wird auch bei diesem Verkehrszeichen kein BSSD-Segment definiert. Im weiteren Verlauf existiert ein Verkehrszeichen zur Festlegung einer zulässigen Höchstgeschwindigkeit von 60 km/h. Aufgrund dessen wird bei diesem Verkehrszeichen das neue Segment 75.0 definiert. Die Straße 0 teilt sich anhand einer Kreuzung in die Straße 1 und die Straße 2 auf. Innerhalb dieser Kreuzung existiert ein Verkehrszeichen zur Festlegung einer zulässigen Höchstgeschwindigkeit von 30 km/h, das in beiden Straßen der Kreuzung Gültigkeit besitzt. Dieses Verkehrszeichen wird über ein `<signal>`-Element in Straße 1 und ein `<signalReference>`-Element in Straße 2 repräsentiert. Entsprechend Regel 7 für die Definition neuer BSSD-Segmente ist somit in beiden Straßen die Definition eines neuen BSSD-Segments erforderlich.

Neben dem OpenDRIVE-Objekt benötigt der zu testende Algorithmus die Datenstrukturen „Befahrbarkeit OpenDRIVE-Fahstreifen“ und „BSSD-Segmente“ als Eingangsdaten. Diese werden manuell durch Betrachtung der Szenerie erzeugt und sind Abbildung 6-21 in Form der `id`-Attribute der OpenDRIVE-Fahstreifen und der blauen BSSD-Segmente zu entnehmen. Dabei repräsentieren `id`-Attribute mit bzw. ohne einen grauen Kasten als Hintergrund befahrbare bzw. nicht-befahrbare OpenDRIVE-Fahstreifen. Das erwartete Ergebnis ist die durch den zu testenden Algorithmus modifizierte Datenstruktur „BSSD-Segmente“. Dies entspricht der Kombination der blauen und gelben BSSD-Segmente in Abbildung 6-21.

## 7 Evaluation

In diesem Kapitel erfolgt eine Evaluation der in dieser Arbeit umgesetzten Methode zur Integration der BSSD in OpenDRIVE. Dazu erfolgt eine Verifikation (Abschnitt 7.1) und Validierung (Abschnitt 7.2) der umgesetzten Methode gegenüber den in Kapitel 4 definierten Detailanforderungen bzw. übergeordneten Anforderungen.

### 7.1 Verifikation

In diesem Abschnitt wird die in dieser Arbeit umgesetzte Methode gegenüber den in Abschnitt 4.1 definierten Detailanforderungen (3.1) bis (3.12), die der Realisierung der übergeordneten Anforderung (3) dienen, verifiziert. Dafür werden die bei der Implementierung definierten Testfälle (siehe Abschnitt 6.4) herangezogen, da diese die Funktionsfähigkeit der umgesetzten Methode entsprechend der untergeordneten Anforderungen (3.1) bis (3.12) nachweisen. Um eine gesamtheitliche Verifikation gegenüber diesen Anforderungen durchzuführen, wird ein Gesamttestfall definiert und implementiert, der eine Kombination aller einzelnen Testfälle repräsentiert. Dazu wird eine Szenerie in *RoadRunner* erstellt, die alle Merkmale der für die einzelnen Testfälle definierten Szenarien enthält.

Eine Übersicht der Szenerie des Gesamttestfalls ist in Abbildung 7-1 dargestellt. Um die Übersichtlichkeit der Abbildung zu gewährleisten, ist nur ein Teilausschnitt der gesamten Szenerie dargestellt, der alle Straßen der Szenerie beinhaltet. Die Szenerie besteht aus vier Straßen, wobei drei dieser Straßen durch eine T-Kreuzung miteinander verbunden sind. Die *id*-Attribute der vier Straßen sind der Abbildung zu entnehmen. Diese werden automatisiert durch *RoadRunner* festgelegt.

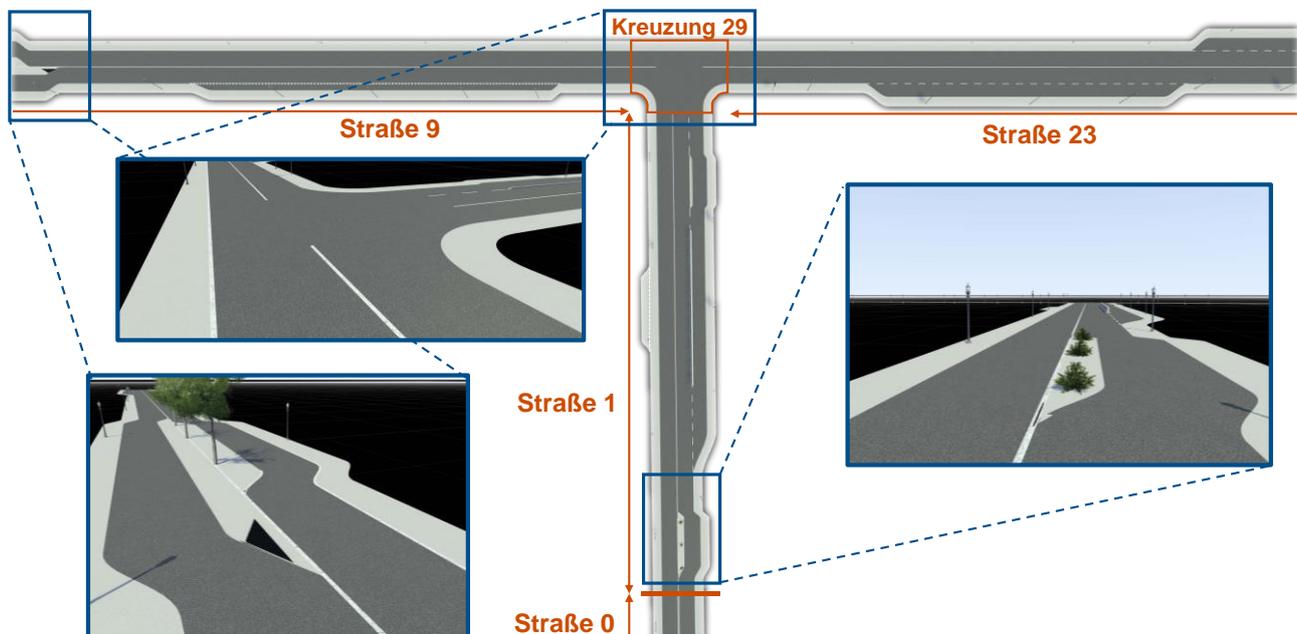


Abbildung 7-1: Übersicht der Szenerie des Gesamttestfalls

Die Zusammensetzung der Szenerie wird im Folgenden exemplarisch für die Straße 9 (links) beschrieben. Die Zusammensetzung der anderen drei Straßen ergibt sich durch ein analoges Vorgehen

und ist anhand vom Anhang A.6 nachzuvollziehen. Die Szenerie der Straße 9 ist übersichtswise in Abbildung 7-2 dargestellt. Um eine übersichtliche Darstellung zu realisieren, ist die Abbildung in drei untereinander angeordnete Abschnitte aufgeteilt. Dabei entspricht der oberste Abschnitt dem linken Teil und der unterste Abschnitt dem rechten Teil von Straße 9. Der Verlauf der Straße von links nach rechts entspricht somit der Betrachtungsrichtung von links nach rechts. Die Abbildung enthält verschiedene Informationsebenen. In Orange ist die Referenzrichtung der Straße angegeben. Die Einteilung der Straße in *lane sections* sowie OpenDRIVE-Fahrstreifen ist in Schwarz dargestellt. Dabei repräsentieren die *id*-Attribute, die einen grauen Kasten als Hintergrund besitzen, befahrbare OpenDRIVE-Fahrstreifen. Folglich sind die *id*-Attribute, die keinen grauen Kasten als Hintergrund besitzen, nicht-befahrbare OpenDRIVE-Fahrstreifen. Die aus der Szenerie resultierende zu erwartende Einteilung der Straße in BSSD-Segmente und BSSD-Fahrstreifen ist in Blau dargestellt. Die in der Szenerie geltenden Verkehrsregeln sind symbolisch über die zugehörigen Verkehrszeichen in der Abbildung repräsentiert. Im Folgenden wird die Szenerie im Kontext der einzelnen Funktionalitäten der Implementierung von links nach rechts beschrieben.

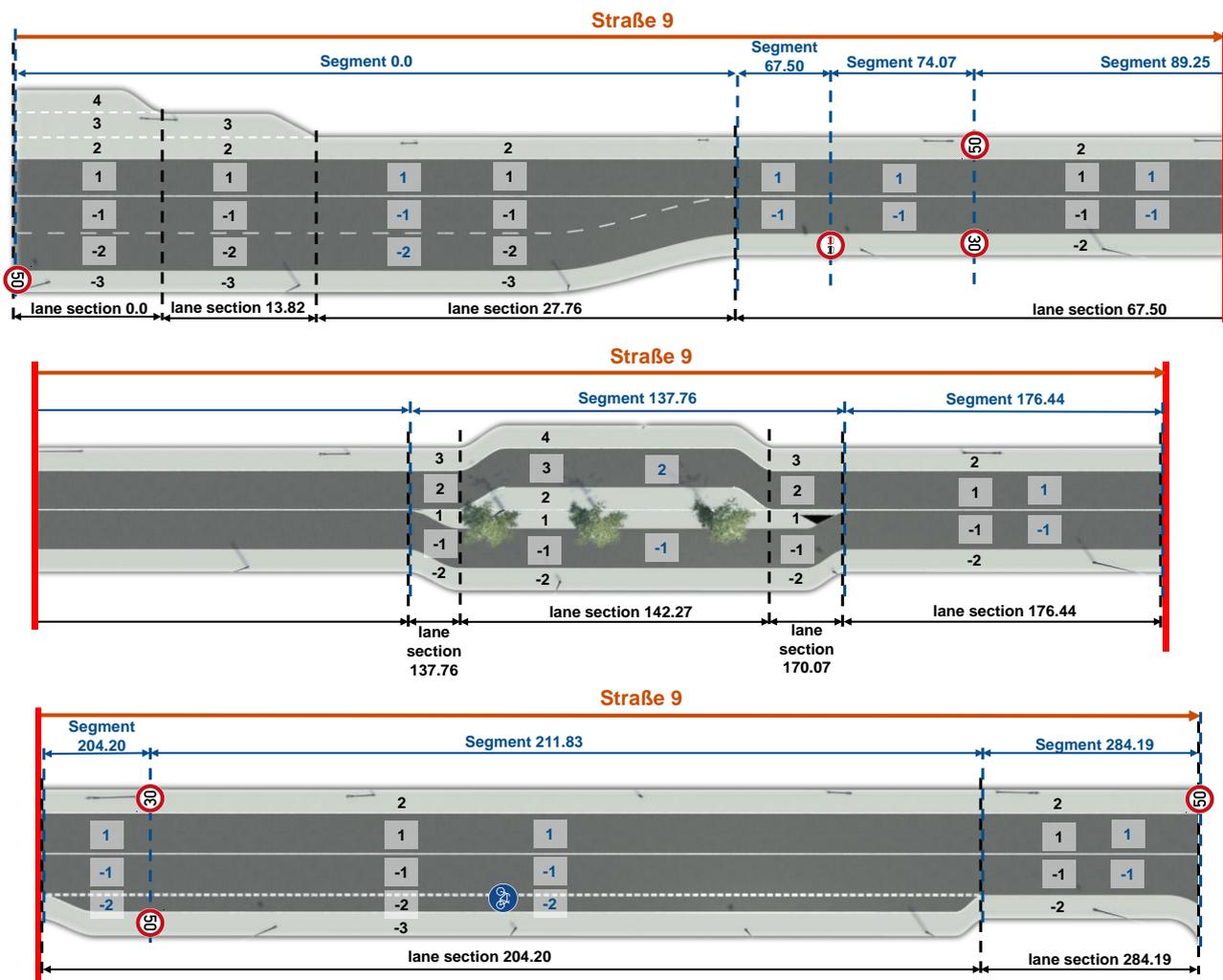


Abbildung 7-2: Szenerie Straße 9, Gesamttestfall

Zu Beginn der Straße ( $s=0.0$ ) existieren drei befahrbare OpenDRIVE-Fahrstreifen, woraus das Segment 0.0 resultiert. Im Bereich dieses BSSD-Segments existieren drei *lane sections*, da sich die

---

Anzahl der OpenDRIVE-Fahrstreifen im oberen Gehweg (OpenDRIVE-Fahrstreifen 2–4) von Drei auf Eins reduziert. Es existieren folglich mehrere *lane sections*, die mit einem BSSD-Segment überschneiden. Dadurch wird die Funktionsfähigkeit von Algorithmus A.5 („Überschneidungen zwischen *lane sections* und Segmenten finden“) sowie Algorithmus A.6 („Suche von zugehörigen OpenDRIVE-Fahrstreifen“) getestet. Bei  $s=67.50$  wird die Anzahl der befahrbaren OpenDRIVE-Fahrstreifen auf Zwei reduziert, was die Funktionsfähigkeit von Regel 2 des Algorithmus A.3 („Automatisierte Extraktion von Segmenten“) testet. Es resultiert das Segment 67.50. Bei  $s=74.07$  existiert ein Überholverbotsschild (Verkehrszeichen 276), das ein Indikationselement repräsentiert. Daraus ergibt sich das Segment 74.07, was die Funktionsfähigkeit von Regel 7 des Algorithmus A.3 testet. Zum Testen der Regel 6 des Algorithmus A.3 wird bei  $s=89.25$  die zulässige Höchstgeschwindigkeit im OpenDRIVE-Fahrstreifen mit der *id* -1 von 50 km/h auf 30 km/h reduziert und im OpenDRIVE-Fahrstreifen mit der *id* 1 von 30 km/h auf 50 km/h erhöht. Aufgrund dessen wird das Segment 74.07 definiert. Dieser Szenerieabschnitt überprüft somit gleichzeitig die Funktionsfähigkeit von Algorithmus A.7 („Automatisierte Extraktion des Verhaltensattributs *Speed Limit*“). Bei  $s=137.76$  wird ein nicht-befahrbarer OpenDRIVE-Fahrstreifen (*id* 1), der eine Verkehrsinsel repräsentiert, in der Mitte der Straße definiert. Daraus resultierend ist die Definition eines neuen BSSD-Segments erforderlich. Dieser Fall wird jedoch von keiner im Algorithmus A.3 implementierten Regeln für die Definition eines neuen BSSD-Segments abgedeckt. Aufgrund dessen wird das Segment 137.76 manuell per Nutzereingabe erstellt, was die Funktionsfähigkeit von Algorithmus A.4 („Manuelles Bearbeiten der befahrbaren Fahrstreifen“) testet. Im Verlauf von Segment 137.76 existieren drei *lane sections*, die sich durch den Beginn und das Ende eines zweiten nicht-befahrbaren OpenDRIVE-Fahrstreifens als Teil der Verkehrsinsel ergeben. Daraus ergibt sich für den BSSD-Fahrstreifen mit der *id* 2 eine Verknüpfung zu drei OpenDRIVE-Fahrstreifen mit unterschiedlichen *id*-Attributen (2, 3, 2), was im Speziellen die Funktionsfähigkeit von Algorithmus A.6 („Suche von zugehörigen OpenDRIVE-Fahrstreifen“) überprüft. Bei  $s=176.44$  beginnt ein Fahrradschutzstreifen, der entsprechend Algorithmus A.2 („Automatisierte Suche von befahrbaren Fahrstreifen“) einen befahrbaren OpenDRIVE-Fahrstreifen repräsentiert und daher die Funktionsfähigkeit dieses Algorithmus überprüft. Bei  $s=204.20$  erfolgt analog zu  $s=89.25$  ein Wechsel der zulässigen Höchstgeschwindigkeit zum Testen von Algorithmus A.7 und Regel 6 des Algorithmus A.3. Beim Ende des Fahrradschutzstreifen ( $s=284.19$ ) wird das letzte BSSD-Segment der Straße 9 definiert, das beim Beginn der Kreuzung 29 endet.

Für die Ausführung dieses Gesamttestfalls wird einerseits die gezeigte in *RoadRunner* erstellte Szenerie als OpenDRIVE-Datei exportiert. Durch die Ausführung der gesamten Implementierung mit dieser OpenDRIVE-Datei ergibt sich eine durch die BSSD-Integration modifizierte OpenDRIVE-Datei. Andererseits wird durch Betrachtung der Szenerie die als Resultat der BSSD-Integration zu erwartende OpenDRIVE-Datei manuell erstellt (siehe Segmentierung in Abbildung 7-2, Abbildung A-3 und Abbildung A-4). Durch den Vergleich dieser beiden Dateien ergibt sich, dass der Gesamttestfall bestanden ist. Daraus resultiert eine Verifikation der in dieser Arbeit umgesetzten Methode zur Integration der BSSD in OpenDRIVE gegenüber den Anforderungen (3.1) bis (3.12).

---

## 7.2 Validierung

In diesem Abschnitt erfolgt eine Validierung der in dieser Arbeit umgesetzten Methode zur Integration der BSSD in OpenDRIVE gegenüber den in Abschnitt 4.1 definierten übergeordneten Anforderungen (1), (2) und (3).

Anforderung (1) fordert, dass die Nutzbarkeit einer OpenDRIVE-Karte durch die BSSD-Integration nicht eingeschränkt werden darf. OpenDRIVE ist ein offener und für jeden nutzbarer Standard, woraus eine Vielzahl von möglichen Einsatzzwecken für eine OpenDRIVE-Karte resultiert. Somit ist eine allgemeingültige Validierung von Anforderung (1) im Rahmen dieser Arbeit nicht durchführbar, da der Nutzungskontext einer OpenDRIVE-Karte nicht eindeutig definiert ist. Da sich die Modifikationen der OpenDRIVE-Karte durch die BSSD-Integration auf das Einfügen eines Wurzelements je Straße beschränken und die bestehenden Elemente und Attribute der OpenDRIVE-Karte nicht verändert werden, ist davon auszugehen, dass die Nutzbarkeit der OpenDRIVE-Karte durch die BSSD-Integration nicht eingeschränkt wird. Für die exemplarische Validierung von Anforderung (1) in einem abgegrenzten Nutzungskontext werden beispielhafte mit der BSSD integrierte OpenDRIVE-Karten in das Programm *RoadRunner* sowie das Online-Tool *OpenDRIVE Viewer*<sup>87</sup>, die jeweils eine Visualisierung von bestehenden OpenDRIVE-Karten ermöglichen, importiert. Dabei sind keine Auffälligkeiten bezüglich einer eingeschränkten Nutzbarkeit der OpenDRIVE-Karten zu beobachten. Für eine weitergehende Validierung der umgesetzten Methode bezüglich Anforderung (1) wäre das Durchführen von Nutzertests in verschiedenen Nutzungskontexten von OpenDRIVE erforderlich.

Anforderung (2) fordert, dass alle in der OpenDRIVE-Karte vor der BSSD-Integration enthaltenen Informationen nach der Integration weiterhin vorhanden sind. Wie bereits erwähnt, nimmt die in dieser Arbeit umgesetzte Methode keine Veränderungen an bestehenden Elementen und Attributen der OpenDRIVE-Karte vor. Aufgrund dessen ist keine Reduktion des in der OpenDRIVE-Karte ursprünglich enthaltenen Informationsgehalts durch die BSSD-Integration möglich. Somit ist die umgesetzte Methode gegenüber Anforderung (2) validiert.

Anforderung (3) fordert, dass die Integration der BSSD in eine OpenDRIVE-Karte alle Informationen der BSSD zu dieser OpenDRIVE-Karte beinhalten muss. Eine allgemeingültige Validierung dieser Anforderung im Rahmen des Einsatzzwecks der BSSD wäre beispielsweise durch den Nachweis der Erfüllung dieser Anforderung für alle in Deutschland vorkommenden Szenarien gegeben. Dies ist im Rahmen dieser Arbeit nicht umsetzbar. Für eine Reduktion des Validierungsaufwands beschreibt Viehof<sup>88</sup> die Methode der Stichprobenvalidität, welche eine Betrachtung mehrerer Konfigurationsstichproben vorsieht. Als ein erster Schritt im Hinblick auf das Erreichen eines stichprobenvaliden Zustands erfolgt im Rahmen dieser Arbeit eine Validierung der umgesetzten Methode für zwei repräsentative real existierende Szenarien – einen Autobahnabschnitt und eine urbane Kreuzung. Dies wird im Folgenden näher beschrieben.

---

<sup>87</sup> Pagel, S.: *OpenDRIVE Viewer* (2022).

<sup>88</sup> Viehof, M.: *Diss., Objektive Qualitätsbewertung von Fahrdynamiksimulationen durch statistische Validierung* (2018), S. 46.

## 7.2.1 Autobahnabschnitt

Für die Validierung eines Autobahnabschnitts wird die von der Firma *3D Mapping Solutions GmbH* im Rahmen des Testfelds für automatisiertes Fahren auf der Bundesautobahn 9 erstellte OpenDRIVE-Karte<sup>89</sup> verwendet. Aus dieser OpenDRIVE-Karte wird ein für eine deutsche Autobahn repräsentativer Ausschnitt ausgewählt. Der ausgewählte Ausschnitt ist ein dreistreifiger Autobahnabschnitt über 2,2 km Länge zwischen der Anschlussstelle Denkendorf und der Anschlussstelle Altmühltal. Der Autobahnabschnitt ist übersichtsweise in Abbildung 7-3 gezeigt. Eine Detailaufnahme ist in Abbildung 7-4 dargestellt. Aufgrund eines Rastplatzes beinhaltet der Autobahnabschnitt eine Autobahnabfahrt und -auffahrt. Weiterhin existieren drei verschiedene zulässige Höchstgeschwindigkeiten (unbegrenzt, 120 km/h, 100 km/h), ein auf Lkw und Pkw mit Anhänger beschränktes Überholverbot sowie weitere nicht für die BSSD relevante Verkehrszeichen, die repräsentativ für eine deutsche Autobahn sind (z.B. Ankündigungstafel, Touristischer Hinweis).

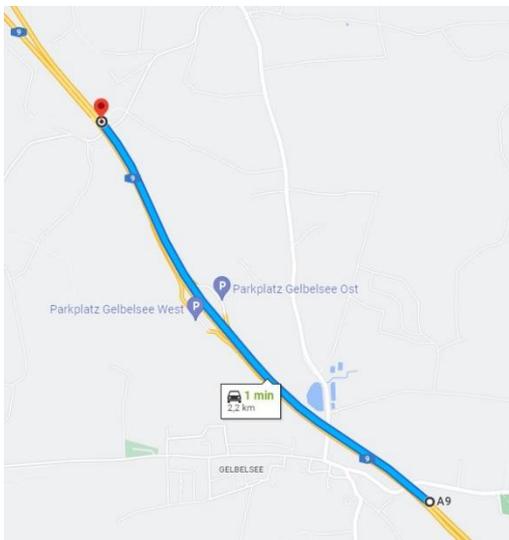


Abbildung 7-3: Übersicht Abschnitt A9<sup>90</sup>



Abbildung 7-4: Detailaufnahme Abschnitt A9<sup>91</sup>

Als Basis für die Validierung des ausgewählten Autobahnabschnitts wird dieser aus der übergeordneten OpenDRIVE-Karte in eine separate OpenDRIVE-Datei ausgelagert. Anhand der parallelen Betrachtung des Autobahnabschnitts mit dem Internetdienst *Mapillary*<sup>91</sup> sowie in *RoadRunner* wird ausgehend von der den Ausschnitt beinhaltenden OpenDRIVE-Karte das zu erwartende Ergebnis der BSSD-Integration erzeugt. Dies erfolgt im Hinblick auf die Funktionsfähigkeit der in dieser Arbeit umgesetzten Methode. Folglich beinhaltet das erwartete Ergebnis die Einteilung der Szenerie in BSSD-Segmente und BSSD-Fahrstreifen sowie die Ausprägungen des Verhaltensattributs *Speed Limit* in den BSSD-Fahrstreifen.

Für die Validierung wird der Vergleich zwischen dem Ergebnis der BSSD-Integration und dem erwarteten Ergebnis in zwei Stufen durchgeführt. In der ersten Stufe wird die BSSD-Integration

<sup>89</sup> BASt: OpenDrive-Kartendaten Testfeld A9 (2017).

<sup>90</sup> Google LLC: Google Maps (2022).

<sup>91</sup> Mapillary AB: Mapillary (2022).

vollkommen automatisiert mit der zu dem Abschnitt zugehörigen OpenDRIVE-Karte ausgeführt (keine Ausführung der Algorithmen A.2 und A.4). In einer zweiten Stufe erfolgt eine vollständige Ausführung der Implementierung inklusive den per Nutzereingabe gesteuerten Algorithmen A.2 und A.4. Aus dem Vergleich zwischen dem Ergebnis der BSSD-Integration und dem erwarteten Ergebnis in den beiden Stufen ergeben sich die in Tabelle 7-1 und Tabelle 7-2 gelisteten Kennzahlen.

Tabelle 7-1: Kennzahlen Stufe 1, Validierung A9

Kennzahl	Wert
Manuell veränderte Klassifizierungen Befahrbarkeit	0
Anteil korrekte Klassifizierung Befahrbarkeit	72 %
Manuell erstellte BSSD-Segmente	0
Manuell entfernte BSSD-Segmente	0
Anteil der korrekten Segmentierung	54 %
Falsch erstellte BSSD-Segmente	14
Anteil extrahierte Ausprägungen <i>Speed Limit</i>	--

Tabelle 7-2: Kennzahlen Stufe 2, Validierung A9

Kennzahl	Wert
Manuell veränderte Klassifizierungen Befahrbarkeit	23
Anteil korrekte Klassifizierung Befahrbarkeit	100 %
Manuell erstellte BSSD-Segmente	11
Manuell entfernte BSSD-Segmente	14
Anteil der korrekten Segmentierung	100 %
Falsch erstellte BSSD-Segmente	0
Anteil extrahierte Ausprägungen <i>Speed Limit</i>	39 %

Bei der ersten Stufe werden entsprechend der vollkommen automatisierten Durchführung der BSSD-Integration keine manuellen Änderungen der Klassifizierung der Befahrbarkeit der OpenDRIVE-Fahrstreifen sowie der gefundenen BSSD-Segmente vorgenommen. Es zeigt sich, dass 72 % aller OpenDRIVE-Fahrstreifen korrekt als befahrbar bzw. nicht-befahrbar klassifiziert werden. Der Anteil der falsch klassifizierten OpenDRIVE-Fahrstreifen ergibt sich durch die Modellierung der OpenDRIVE-Fahrstreifen, die den linken Randstreifen der Autobahn repräsentieren, mit dem Attribut `type="shoulder"`. Diese Ausprägung wurde bei der Implementierung als nicht-befahrbar klassifiziert (siehe Tabelle A-5), da diese nach Definition der Spezifikation von OpenDRIVE einen unbefestigten Bereich neben der Straße repräsentiert. Entsprechend wäre in diesem Fall eine Modellierung dieser OpenDRIVE-Fahrstreifen mit dem Attribut `type="border"` passender, woraus eine korrekte Klassifizierung aller OpenDRIVE-Fahrstreifen als befahrbar bzw. nicht-befahrbar resultieren würde.

Weiterhin werden 54 % aller zu erwartenden BSSD-Segmente automatisiert gefunden. Die nicht automatisiert gefundenen BSSD-Segmente ergeben sich hauptsächlich durch Änderungen des Verhaltensattributs *Boundary* in Bezug auf die lateralen Begrenzungen der BSSD-Fahrstreifen. Zwei beispielhafte Ausschnitte, bei denen ein BSSD-Segment nicht automatisiert gefunden wird, sind in Abbildung 7-5 und Abbildung 7-6 dargestellt. Im ersten Beispiel ändert sich die Fahrstreifenmarkierung zwischen den BSSD-Fahrstreifen mit der `id -3` und `-4` von durchgezogen auf gestrichelt, woraus sich eine Änderung des dem Verhaltensattribut *Boundary* zugehörigen Attributs `crossing` von `"prohibited"` zu `"allowed"` ergibt. Im zweiten Beispiel endet die rechts des BSSD-Fahrstreifens mit der `id -4` befindliche Leitplanke, sodass sich das Attribut `crossing` von `"not possible"` zu `"prohibited"` ändert.



Abbildung 7-5: Segmentierung A9, Beispiel 1<sup>92</sup>



Abbildung 7-6: Segmentierung A9, Beispiel 2<sup>92</sup>

Darüber hinaus ergeben sich durch die Ausführung der BSSD-Integration in der ersten Stufe 14 falsch erstellte BSSD-Segmente, deren Ursache die Segmentierung basierend auf Verkehrszeichen ist. In dem Autobahnabschnitt existieren mehrere Verkehrszeichen der Verkehrszeichen 274 (zulässige Höchstgeschwindigkeit) und 276 (Überholverbot). Da diese Indikationselemente repräsentieren, werden entsprechend Regel 7 (Algorithmus A.3) BSSD-Segmente bei diesen Verkehrszeichen definiert. Beim Verkehrszeichens 274 ist in diesem Anwendungsfall aus zwei Gründen die Definition eines BSSD-Segments falsch. Einerseits, weil die Segmentierung basierend auf der Änderung der zulässigen Höchstgeschwindigkeit bereits über das in den OpenDRIVE-Fahrstreifen enthaltene `<speed>`-Element abgedeckt wird (siehe Regel 6, Algorithmus A.3). Da die `s`-Koordinaten der `<speed>`-Elemente jeweils geringfügig von denen der `<signal>`-Elemente, die dem Verkehrszeichen zugehörig sind, abweichen, werden falsche BSSD-Segmente definiert. Andererseits werden die Verkehrszeichen 274 in dem Autobahnabschnitt mehrfach wiederholt, sodass keine Änderung des Verhaltensattributs *Speed Limit* resultiert. In diesen Fällen ist somit die Definition eines BSSD-Segments ebenfalls falsch. Die Segmentierung beim Verkehrszeichen 276 ist in diesem Anwendungsfall falsch, weil das durch das Verkehrszeichen definierte Überholverbot durch ein Zusatzschild auf Lkw und Pkw mit Anhänger beschränkt ist. Da im Kontext der Projekte PRORETA 5 und UNICARagil diese Fahrzeugtypen nicht relevant sind, ist eine Segmentierung nicht notwendig. Wäre das aus BSSD-Sicht betrachtete Ego-Fahrzeug beispielsweise ein automatisierter Lkw, wäre die Definition eines BSSD-Segments in diesem Anwendungsfall korrekt.

Die Angabe des Anteils der extrahierten Ausprägungen des Verhaltensattributs *Speed Limit* ist bei der ersten Stufe nicht zweckmäßig, da die Segmentierung und somit die BSSD-Fahrstreifen in dieser Stufe nicht vollständig und korrekt sind.

Bei der zweiten Stufe werden die zuvor angesprochenen OpenDRIVE-Fahrstreifen mit dem Attribut `type="shoulder"` manuell als befahrbar klassifiziert (Algorithmus A.2). Daraus resultiert, dass alle OpenDRIVE-Fahrstreifen korrekt als befahrbar bzw. nicht-befahrbar klassifiziert sind. Darüber hinaus werden die von der BSSD-Integration nicht automatisiert extrahierten BSSD-Segmente

<sup>92</sup> Mit Änderungen entnommen von: Mapillary AB: Mapillary (2022).

---

manuell erstellt und die durch die Verkehrszeichen falsch extrahierten BSSD-Segmente manuell entfernt. Somit ergibt sich eine vollständige und korrekte Segmentierung des betrachteten Autobahnabschnitts.

Der Anteil der extrahierten Ausprägungen des Verhaltensattributs *Speed Limit* (entlang und entgegen der Fahrtrichtung) beträgt 39 %. Der umgekehrte Anteil der Ausprägungen wird folglich nicht automatisiert aus der OpenDRIVE-Karte bestimmt. Die Ursache dafür ist einerseits, dass nur für die drei Hauptfahrstreifen der Autobahn das Element `<speed>` in der OpenDRIVE-Karte definiert ist. Aufgrund dessen wird nur in den zu diesen OpenDRIVE-Fahrstreifen zugehörigen BSSD-Fahrstreifen *Speed Limit* extrahiert. Andererseits sind die OpenDRIVE-Fahrstreifen, die den linken Randstreifen repräsentieren, fälschlicherweise mit der `id 1` und somit mit einem anderen Vorzeichen als die `id`-Attribute der restlichen OpenDRIVE-Fahrstreifen definiert<sup>93</sup>. Aufgrund dessen wird die Straße bei der Ausführung des Algorithmus A.7 nicht als Einbahnstraße erkannt und die Ausprägungen von *Speed Limit* der restlichen BSSD-Fahrstreifen entgegen der vorgesehenen Fahrtrichtung basieren somit auf den OpenDRIVE-Fahrstreifen mit der `id 1`. Da in diesen OpenDRIVE-Fahrstreifen jedoch kein `<speed>`-Element definiert ist, ist es nicht möglich, für die restlichen OpenDRIVE-Fahrstreifen die Ausprägung von *Speed Limit* entgegen der vorgesehenen Fahrtrichtung zu bestimmen. Eine vollständige Bestimmung des Verhaltensattributs *Speed Limit* durch die umgesetzte Methode wäre möglich, wenn in der OpenDRIVE-Karte für alle befahrbaren OpenDRIVE-Fahrstreifen die zulässige Höchstgeschwindigkeit im Element `<speed>` definiert wird.

### 7.2.2 Urbane Kreuzung

Für die Validierung einer urbanen Kreuzung wird eine von der Firma *3D Mapping Solutions GmbH* erstellte OpenDRIVE-Karte, die einen Auszug eines Wohngebiets der Mittelstadt Griesheim enthält, verwendet. Aus dieser OpenDRIVE-Karte wird ein Ausschnitt ausgewählt, der für eine in einem städtischen Wohngebiet befindliche Kreuzung repräsentativ ist. Der ausgewählte Ausschnitt wird durch die Kreuzung Lilienthal-/Groenhoffstraße sowie die Abschnitte dieser Straßen kurz vor der Kreuzung gebildet. Der Ausschnitt ist übersichtsweise in Abbildung 7-7 rot markiert. Eine Detailaufnahme ist in Abbildung 7-8 gezeigt. Das Straßenlayout des Ausschnitts ist repräsentativ für ein Wohngebiet und besteht aus einer Zweirichtungsstraße, einem Gehweg je Seite sowie Park- und Sperrflächen. Weiterhin existieren verschiedene Verkehrszeichen sowie mehrere Grünflächen und Bäume.

Das Vorgehen zur Validierung der ausgewählten Kreuzung ist analog zum Vorgehen der Validierung des Autobahnabschnitts (siehe Abschnitt 7.2.1). Der Vergleich zwischen dem Ergebnis der BSSD-Integration und dem erwarteten Ergebnis erfolgt in gleicher Weise in zwei Stufen. In der ersten Stufe wird die BSSD-Integration vollkommen automatisiert ausgeführt, während in der zweiten Stufe eine vollständige Ausführung inklusive der die manuelle Nutzereingaben beinhaltenden Algorithmen A.2 und A.4 erfolgt.

---

<sup>93</sup> Das andere Vorzeichen des `id`-Attributs repräsentiert einen OpenDRIVE-Fahrstreifen mit entgegengesetzter Fahrtrichtung

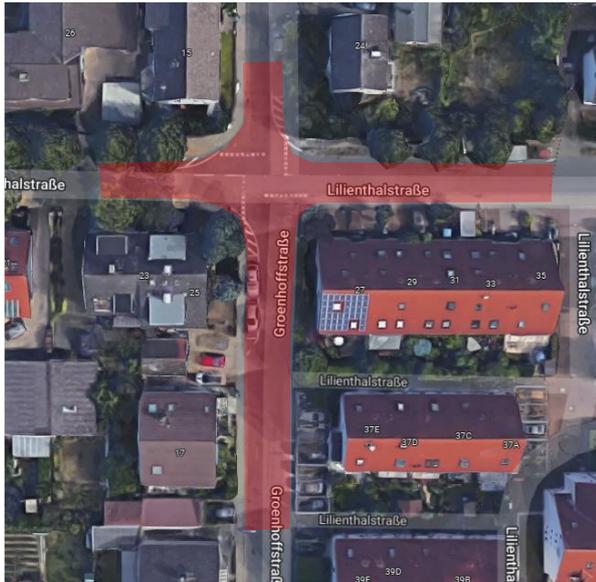


Abbildung 7-7: Übersicht Ausschnitt Griesheim<sup>94</sup>



Abbildung 7-8: Detailaufnahme Ausschnitt Griesheim<sup>95</sup>

Aus dem Vergleich zwischen dem Ergebnis der BSSD-Integration und dem erwarteten Ergebnis in den beiden Stufen ergeben sich die in Tabelle 7-3 und Tabelle 7-4 gelisteten Kennzahlen.

Tabelle 7-3: Kennzahlen Stufe 1, Validierung Griesheim

Kennzahl	Wert
Manuell veränderte Klassifizierungen Befahrbarkeit	0
Anteil korrekte Klassifizierung Befahrbarkeit	78 %
Manuell erstellte BSSD-Segmente	0
Manuell entfernte BSSD-Segmente	0
Anteil der korrekten Segmentierung	84 %
Falsch erstellte BSSD-Segmente	5
Anteil extrahierte Ausprägungen <i>Speed Limit</i>	--

Tabelle 7-4: Kennzahlen Stufe 2, Validierung Griesheim

Kennzahl	Wert
Manuell veränderte Klassifizierungen Befahrbarkeit	58
Anteil korrekte Klassifizierung Befahrbarkeit	100 %
Manuell erstellte BSSD-Segmente	3
Manuell entfernte BSSD-Segmente	5
Anteil der korrekten Segmentierung	100 %
Falsch erstellte BSSD-Segmente	0
Anteil extrahierte Ausprägungen <i>Speed Limit</i>	46 %

In der ersten Stufe werden 78 % aller OpenDRIVE-Fahrstreifen korrekt als befahrbar bzw. nicht-befahrbar klassifiziert. Der Anteil der falsch klassifizierten OpenDRIVE-Fahrstreifen ergibt sich durch die Modellierung des Bordsteins der Gehwege mit dem Attribut `type="border"`. Diese Ausprägung des `type`-Attributs wurde bei der Implementierung als repräsentativ für einen befahrbaren OpenDRIVE-Fahrstreifen klassifiziert (siehe Tabelle A-5). Der linke und rechte Randstreifen der Zweirichtungsfahrstreifen werden ebenfalls mit dem Attribut `type="border"` modelliert. Dadurch ergibt sich für diese Szenerie, dass zu einer Ausprägung des `type`-Attributs sowohl befahrbare als auch nicht-befahrbare OpenDRIVE-Fahrstreifen zugehörig sind. Die Modellierung der OpenDRIVE-

<sup>94</sup> Mit Änderungen entnommen von: Google LLC: Google Maps (2022).

<sup>95</sup> Mapillary AB: Mapillary (2022).

Fahrstreifen mit dem `type`-Attribut ist exemplarisch in Abbildung 7-9 für einen Ausschnitt der Szenerie dargestellt. Für einen Bordstein wäre die Modellierung mit dem Attribut `type="curb"` entsprechend der OpenDRIVE-Spezifikation passender. In diesem Fall würden alle OpenDRIVE-Fahrstreifen automatisiert korrekt als befahrbar bzw. nicht-befahrbar klassifiziert werden.

Darüber hinaus werden 84 % aller zu erwartenden BSSD-Segmente automatisiert gefunden. Der Anteil der nicht gefundenen BSSD-Segmente ergibt sich ausschließlich durch laterale Begrenzungen von BSSD-Fahrstreifen im Kontext des Verhaltensattributs *Boundary*. Ein beispielhafter Ausschnitt, bei dem ein BSSD-Segment nicht automatisiert gefunden wird, ist in Abbildung 7-10 gezeigt. In diesem Fall ändert sich die Ausprägung des Attributs `crossing` (Verhaltensattribut *Boundary*) in Folge des Wechsels der Parkfläche zu einer Sperrfläche von "conditional" zu "prohibited". Die restlichen nicht automatisiert gefundenen BSSD-Segmente ergeben sich durch die zu Abbildung 7-10 äquivalenten Situation in den anderen Straßen.



Abbildung 7-9: Beispiel `type`-Attribut, Griesheim<sup>96</sup>



Abbildung 7-10: Beispiel Segmentierung, Griesheim<sup>96</sup>

Weiterhin resultieren aus der BSSD-Integration in der ersten Stufe fünf falsch erstellte BSSD-Segmente, deren Ursache analog zum Autobahnabschnitt die Segmentierung basierend auf Verkehrszeichen ist. Einerseits existieren mehrere dynamische `<signal>`-Elemente, bei denen entsprechend Regel 8 (Algorithmus A.3) BSSD-Segmente definiert werden. Die reale Repräsentation dieser dynamischen `<signal>`-Elemente war trotz mehrfacher und unterschiedlicher Betrachtungen der Szenerie nicht auffindbar. Andererseits existiert auf einem Gehweg innerhalb der betrachteten Kreuzung das Verkehrszeichen 239 („Gehweg“). Entsprechend der Klassifizierung der Verkehrszeichen des VzKat von Glatzki et al.<sup>97</sup> repräsentiert dieses Verkehrszeichen ein Indikationselement. Folglich wird entsprechend Regel 7 (Algorithmus A.3) bei dem `<signal>`-Element, das dieses Verkehrszeichen repräsentiert, sowie den dazugehörigen `<signalReference>`-Elementen ein BSSD-Segment definiert. In diesem Fall bezieht sich die Gültigkeit des Verkehrszeichens 239 jedoch auf den Gehweg

<sup>96</sup> Mit Änderungen entnommen von: Mapillary AB: Mapillary (2022).

<sup>97</sup> Glatzki, F.; Winner, H.: Inferenz von Verhaltensattributen der verhaltenssemantischen Szeneriebeschreibung (2022), S. 4.

---

und somit einen nicht-befahrbaren OpenDRIVE-Fahstreifen. Aufgrund dessen ist in diesem Anwendungsfall keine Segmentierung bei diesem Verkehrszeichen erforderlich. Analog zum Autobahnabschnitt ist die Angabe des Anteils der extrahierten Ausprägungen des Verhaltensattributs *Speed Limit* in der ersten Stufe nicht zweckmäßig.

Bei der Ausführung der BSSD-Integration in der zweiten Stufe werden alle OpenDRIVE-Fahstreifen, die einen Bordstein repräsentieren, manuell als nicht-befahrbar klassifiziert. Daraus ergibt sich, dass die Befahrbarkeit aller OpenDRIVE-Fahstreifen korrekt klassifiziert wird. Weiterhin werden die nicht automatisiert gefundenen BSSD-Segmente manuell erstellt sowie die falsch gefundenen BSSD-Segmente manuell entfernt. Somit resultiert eine vollständige und korrekte Segmentierung der betrachteten urbanen Kreuzung.

Es werden 46 % der Ausprägungen des Verhaltensattributs *Speed Limit* bestimmt. Die Ursache für den Anteil der nicht gefundenen Ausprägungen ergibt sich analog zum zuvor betrachteten Autobahnabschnitt, da das `<speed>`-Element nur in den OpenDRIVE-Fahstreifen, die den Zweirichtungsfahstreifen repräsentieren, definiert ist. Eine vollständige Bestimmung des Verhaltensattributs *Speed Limit* wäre möglich, wenn für alle befahrbaren OpenDRIVE-Fahstreifen die zulässige Höchstgeschwindigkeit im Element `<speed>` hinterlegt wird.

Zusammenfassend ergibt sich, dass die in dieser Arbeit umgesetzte Methode zur Integration der BSSD in OpenDRIVE im Kontext einer vollständigen und korrekten Segmentierung für einen repräsentativen Autobahnabschnitt und eine repräsentative urbane Kreuzung valide in Bezug auf Anforderung (3) ist. Durch manuelles Eintragen der Ausprägungen aller Verhaltensattribute ist es bei diesen beiden Szenarien möglich, eine vollständig mit BSSD-Informationen integrierte OpenDRIVE-Karte zu erzeugen.

---

## 8 Gesamtbewertung

---

In diesem Kapitel wird eine gesamtheitliche Bewertung basierend auf den Erkenntnissen dieser Arbeit vorgenommen. Dazu wird in Abschnitt 8.1 die allgemeine Integrierbarkeit von BSSD in OpenDRIVE bewertet. Die in dieser Arbeit umgesetzte Methode zur Integration der BSSD in OpenDRIVE wird in Abschnitt 8.2 bewertet.

### 8.1 Bewertung der Integrierbarkeit von BSSD in OpenDRIVE

Unabhängig von der BSSD ist die Erweiterbarkeit von OpenDRIVE grundsätzlich als gut zu beurteilen. Mit dem Element `<userData>` (siehe Abschnitt 3.3.3) ermöglicht OpenDRIVE die Ergänzung von benutzerdefinierten Strukturen mit beliebiger Strukturtiefe, ohne dabei eine Änderung der bestehenden Struktur vorzunehmen. Somit ergibt sich eine Anpassbarkeit an spezielle Anwendungsfälle, ohne die allgemeine Verwendbarkeit und Austauschbarkeit der OpenDRIVE-Karten einzuschränken. Dies zeigt sich auch daran, dass die in dieser Arbeit abgeleiteten Konzepte (siehe Abschnitt 5.1) zum Großteil bzw. vollständig auf der Verwendung des `<userData>`-Elements basieren.

Bei Betrachtung der Eigenschaften und des Aufbaus von OpenDRIVE aus Sicht der BSSD zeigt sich, dass OpenDRIVE gute Grundvoraussetzungen für eine Integration der BSSD bietet. Die primäre Ursache dafür ist, dass die Eigenschaften des minimalen Elements des Straßennetzwerks einer OpenDRIVE-Karte, dem OpenDRIVE-`<lane>`-Element, passend zu den Eigenschaften des minimalen Elements der BSSD, dem atomaren Verhaltensraum, sind. Ein atomarer Verhaltensraum entspricht im Großteil der Anwendungsfälle einem in longitudinaler Richtung abgegrenzten Teilstück eines Fahrstreifens. Somit besitzt dieser klar definierte laterale sowie longitudinale Begrenzungen. Diese Eigenschaften treffen in gleicher Weise auf einen OpenDRIVE-Fahrstreifen zu. Somit ist es naheliegend, diese in OpenDRIVE bereits existierende Struktur als Referenz für das Einfügen des Verhaltensraums der BSSD zu nutzen.

Mit diesen Grundvoraussetzungen war es in dieser Arbeit möglich, drei verschiedene Konzepte zu entwickeln, die eine Integration der BSSD in OpenDRIVE prinzipiell ermöglichen. Das umgesetzte Konzept erreicht dies ohne die ursprünglich in der OpenDRIVE-Karte vorhandenen Informationen zu verändern. Dabei existieren einige Einschränkungen, die im Folgenden beschrieben werden. Die atomaren Verhaltensräume werden bei dem umgesetzten Konzept durch eine Referenzierung auf einen oder mehrere OpenDRIVE-Fahrstreifen definiert. Da es erforderlich ist, dass die referenzierten OpenDRIVE-Fahrstreifen innerhalb einer OpenDRIVE-Straße definiert sind, ergibt sich eine Beschränkung für die longitudinale Begrenzung des atomaren Verhaltensraums. Es ist erforderlich, dass diese innerhalb einer Straße der OpenDRIVE-Karte definiert ist. Es ist davon auszugehen, dass im Großteil der Anwendungsfälle beim Beginn bzw. Ende einer Straße der OpenDRIVE-Karte ebenfalls die longitudinale Begrenzung eines atomaren Verhaltensraums zu definieren ist. Da keine klaren Regeln für die Definition des Beginns bzw. Endes einer OpenDRIVE-Straße existieren, ist dennoch denkbar, dass durch diese Beschränkung in manchen Anwendungsfällen die Definition von redundanten atomaren Verhaltensräumen erforderlich ist. Es ist jedoch davon auszugehen, dass sich

---

dadurch kein Nachteil bei möglichen praktischen Anwendungen der mit der BSSD integrierten OpenDRIVE-Karte ergibt.

Eine weitere Einschränkung ergibt sich in Bezug auf die lateralen Begrenzungen eines atomaren Verhaltensraums. Im Unterschied zu der longitudinalen Begrenzung sind die lateralen Begrenzungen eines atomaren Verhaltensraums nicht veränderbar, da das umgesetzte Konzept die bestehende laterale Einteilung der Straße in OpenDRIVE-Fahrsstreifen übernimmt. Auf Basis dieser Einteilung ist erforderlich zu entscheiden, ob ein OpenDRIVE-Fahrsstreifen mit der BSSD modelliert wird oder nicht. Bei einer korrekten Modellierung der OpenDRIVE-Karte ist die laterale Einteilung der Straße jedoch passend zu den lateralen Begrenzungen der atomaren Verhaltensräume. Somit ist nicht davon auszugehen, dass diese Einschränkung einen relevanten Nachteil darstellt.

Darüber hinaus ergibt sich in Bezug auf die Auswahl der zu verarbeitenden OpenDRIVE-Karten eine weitere Einschränkung, die aus der Implementierung der Methode resultiert. Bei der Implementierung wurde aus mehreren Gründen der Sonderfall von einseitig definierten *lane sections* nicht berücksichtigt (siehe Abschnitt 6.3.5.1). Somit ist keine Integration der BSSD in OpenDRIVE-Karten möglich, die über dieses Element verfügen. Da jedoch davon auszugehen ist, dass dieser Sonderfall in der Praxis nicht relevant ist (siehe Abschnitt 6.3.5.1), stellt diese Einschränkung keinen Nachteil dar.

Zusammenfassend lässt sich festhalten, dass die Eigenschaften von OpenDRIVE eine vollständige und korrekte Integration der BSSD durch die Ergänzung einer benutzerdefinierten Struktur ermöglichen. Aufgrund der dabei erfolgenden Referenzierung auf Strukturen von OpenDRIVE sind bestimmte Einschränkungen zu beachten. Bei der in dieser Arbeit umgesetzten Methode ist davon auszugehen, dass aus diesen Einschränkungen kein in der Praxis relevanter Nachteil resultiert.

## 8.2 Bewertung der umgesetzten Methode

Die im Rahmen dieser Arbeit umgesetzte Methode zur Integration der BSSD in OpenDRIVE ermöglicht für eine bestehende OpenDRIVE-Karte das Einfügen einer vollständigen und korrekten Segmentierung sowie, in Abhängigkeit des Informationsgehalts der Karte, die automatisierte Extraktion des Verhaltensattributs *Speed Limit*. Dies wurde exemplarisch anhand von zwei realen repräsentativen Szenarien demonstriert. Die in der OpenDRIVE-Karte enthaltenen Informationen erlauben, dass die Ausführung der Methode grundsätzlich automatisiert erfolgt. Jedoch bietet die Methode an zwei Stellen die Möglichkeit des manuellen Bearbeitens bzw. Ergänzens der automatisierten Ergebnisse. Wie sich anhand der Validierung der realen Szenarien gezeigt hat, ist davon auszugehen, dass im Großteil der Anwendungsfälle die Ausführung dieser beiden manuell durchzuführenden Schritte notwendig ist, um ein vollständiges und korrektes Ergebnis zu erreichen. Im Folgenden werden die Funktionalitäten der umgesetzten Methode, die für eine korrekte BSSD-Integration am wichtigsten sind, näher bewertet.

---

## 8.2.1 Klassifizierung der Befahrbarkeit von OpenDRIVE-Fahrstreifen

Die erste Stelle, bei der die umgesetzte Methode die Möglichkeit des manuellen Bearbeitens der automatisierten Ergebnisse bietet, ist die Klassifizierung der OpenDRIVE-Fahrstreifen als *befahrbar* bzw. *nicht-befahrbar*. Diese Klassifizierung ist erforderlich, da OpenDRIVE neben der eigentlichen Fahrbahn, die für die BSSD relevant ist, weitere Teile des Straßennetzwerks, wie beispielsweise Gehwege, einheitlich über OpenDRIVE-Fahrstreifen modelliert. Folglich ist für jeden OpenDRIVE-Fahrstreifen erforderlich zu entscheiden, ob dieser mit der BSSD modelliert wird oder nicht. Bei der umgesetzten Methode wird die Fahrbahnzugehörigkeit anhand des zu einem OpenDRIVE-Fahrstreifen zugehörigen Attributs `type` bewertet, das die Art eines OpenDRIVE-Fahrstreifens über vordefinierte Ausprägungen beschreibt. Bei der Validierung der Methode anhand repräsentativer realer Szenarien sowie der Betrachtung weiterer realer Szenarien im Kontext der Verwendung des `type`-Attributs hat sich gezeigt, dass anhand dieses Attributs die Zugehörigkeit zur Fahrbahn im Großteil der Fälle korrekt erkannt wird. So liegt der Anteil der korrekt klassifizierten OpenDRIVE-Fahrstreifen beim betrachteten Autobahnabschnitt bei 72 % (siehe Tabelle 7-1) sowie bei der urbanen Kreuzung bei 78 % (siehe Tabelle 7-3). Die Ursache für die falsch klassifizierten OpenDRIVE-Fahrstreifen ist die uneinheitliche Verwendung einzelner Ausprägungen des `type`-Attributs in Bezug auf die Fahrbahnzugehörigkeit. So werden insbesondere die Ausprägungen "shoulder", "border" (unbefestigter/befestigter Randstreifen) und "biking" (Fahrradweg) sowohl für auf der Fahrbahn befindliche als auch für neben der Fahrbahn befindliche OpenDRIVE-Fahrstreifen verwendet (siehe beispielsweise Abbildung 6-20 und Abbildung 7-9). Aufgrund dessen und da eine korrekte Verwendung des `type`-Attributs in allen OpenDRIVE-Karten nicht sichergestellt ist, bietet die umgesetzte Methode die Möglichkeit des manuellen Bearbeitens der Klassifizierung der Befahrbarkeit der OpenDRIVE-Fahrstreifen. Um den Anteil der korrekt erfolgenden Klassifizierung der Fahrbahnzugehörigkeit zu erhöhen, sind zwei Lösungsansätze denkbar. Einerseits wäre es möglich, für jeden OpenDRIVE-Fahrstreifen ein zusätzliches verpflichtendes Attribut zu definieren, das anhand einer binären Ausprägung festlegt, ob dieser OpenDRIVE-Fahrstreifen einen Teil der Fahrbahn darstellt oder nicht. Andererseits bestände die Möglichkeit, weitere Ausprägungen des `type`-Attributs zu definieren sowie die Beschreibung der aktuell existierenden Ausprägungen in Bezug auf die Fahrbahnzugehörigkeit zu präzisieren. Somit wäre eine Verbesserung der Klassifizierung der Fahrbahnzugehörigkeit der OpenDRIVE-Fahrstreifen anhand des `type`-Attributs möglich.

## 8.2.2 Segmentierung

Die zweite Stelle, bei der die umgesetzte Methode die Möglichkeit des manuellen Bearbeitens der automatisierten Ergebnisse bietet, ist das Erstellen bzw. Löschen von BSSD-Segmenten. Die Einteilung der in der OpenDRIVE-Karte enthaltenen Szenerie in BSSD-Segmente ist abhängig vom Zustand der Verhaltensattribute. Da der Zustand der Verhaltensattribute im Allgemeinen nicht bekannt ist, ist erforderlich, dass die BSSD-Segmente manuell erstellt werden. Bei der in dieser Arbeit umgesetzten Methode werden dennoch anhand von BSSD-relevanten Informationen, die in der OpenDRIVE-Karte enthalten sind, bestimmte BSSD-Segmente automatisiert extrahiert. Die automatisierte Extraktion von BSSD-Segmenten basiert auf Regeln, die bei der Verarbeitung der

---

OpenDRIVE-Karte geprüft werden (siehe Abschnitt 6.3.3.1). Diese Regeln für das Finden von BSSD-Segmenten beziehen auf Änderungen im Straßenlayout, Geschwindigkeitsbegrenzungen sowie Verkehrszeichen. Bei der Validierung der repräsentativen Szenarien war es somit möglich 54 % (Autobahnabschnitt, siehe Tabelle 7-1) bzw. 84 % (urbane Kreuzung, siehe Tabelle 7-3) aller BSSD-Segmente automatisiert zu finden. Dabei hat sich gezeigt, dass, mit Ausnahme der Segmentierung basierend auf Verkehrszeichen, alle durch die definierten Regeln automatisiert extrahierten BSSD-Segmente korrekt sind.

Die Segmentierung basierend auf Verkehrszeichen wird ausschließlich anhand der Betrachtung der Regel des Verkehrszeichens vorgenommen. Aus dieser Betrachtungsweise ergeben sich aus mehreren Gründen Fehler. Einerseits wird der Zustand der Verhaltensattribute vor dem Verkehrszeichen nicht beachtet. Somit wird auch bei Verkehrszeichen segmentiert, bei denen eine bereits bestehende Regel nur wiederholt wird. Wie die Validierung gezeigt hat, tritt dieses Problem beispielweise auf Autobahnen auf. Andererseits werden mögliche Zusatzzeichen, welche die Bedeutung eines Verkehrszeichens konkretisieren, nicht berücksichtigt. So wurde beispielsweise bei dem betrachteten Autobahnabschnitt bei einem Überholverbot, das auf Lkw beschränkt ist, fälschlicherweise segmentiert. Weiterhin wird der Gültigkeitsbereich der Verkehrszeichen nicht betrachtet. So hat sich bei der Validierung der urbanen Kreuzung gezeigt, dass fälschlicherweise bei einem Verkehrszeichen „Gehweg“, das sich nur auf den Bürgersteig und somit einen nicht-befahrbaren OpenDRIVE-Fahrstreifen bezieht, segmentiert wird. Dieses Verkehrszeichen repräsentiert grundsätzlich ein Indikationselement, ist jedoch in diesem Anwendungsfall aufgrund der Gültigkeit für einen nicht-befahrbaren OpenDRIVE-Fahrstreifen nicht für die BSSD relevant. Für eine korrekte automatisierte Segmentierung basierend auf Verkehrszeichen ist folglich eine deutlich detailliertere Betrachtung bezüglich der vor dem Verkehrszeichen geltenden Regeln, der eventuell vorhandenen Einschränkungen des Verkehrszeichens sowie der von dem Verkehrszeichen betroffenen Fahrstreifen erforderlich.

Zur Reduktion des Aufwands für das manuelle Erstellen der BSSD-Segmente ist eine weitere Verbesserung der in dieser Arbeit umgesetzten automatisierten Extraktion von BSSD-Segmenten möglich. Dafür sind weitere in der OpenDRIVE-Karte vorhandenen Informationen verwendbar. Bei der Validierung der realen Szenarien hat sich gezeigt, dass vorwiegend BSSD-Segmente, die sich durch eine Änderung des Verhaltensattributs *Boundary* im Kontext der lateralen Begrenzung eines BSSD-Fahrstreifens ergeben, nicht automatisiert gefunden werden. Es erscheint mit begrenztem Aufwand möglich, eine Vielzahl der dabei auftretenden Fälle durch eine Segmentierung basierend auf den in der OpenDRIVE-Karte definierten Fahrstreifenmarkierungen automatisiert zu finden. Weiterhin ist im Kontext des Verhaltensattributs *Boundary* eine Nutzung der im Rahmen des Algorithmus A.7 („Automatisierte Extraktion des Verhaltensattributs *Speed Limit*“) extrahierten Informationen bezüglich der Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen für das automatisierte Extrahieren von BSSD-Segmenten möglich. Auch die in diesem Kontext gesammelten Daten über die in der OpenDRIVE-Karte enthaltenen Objekte sind für eine Segmentierung nutzbar. Dabei sind jedoch einige Annahmen und Ungenauigkeiten zu beachten, die bei der Beurteilung der Extraktion des Verhaltensattributs *Speed Limit* beschrieben werden (siehe Abschnitt 8.2.3). Darüber hinaus erscheint im

---

Kontext des Verhaltensattributs *Overtake* eine Segmentierung basierend auf den Verkehrszeichen und Fahrstreifenmarkierungen der OpenDRIVE-Karte umsetzbar.

Zusammenfassend ergibt sich, dass die in dieser Arbeit umgesetzte Methode eine vollständige und korrekte Segmentierung einer bestehenden OpenDRIVE-Karte erreicht. Die dabei erfolgende automatisierte Extraktion von BSSD-Segmenten funktioniert zum Großteil korrekt und zeigt somit das Potential für die Extraktion von BSSD-Segmenten aus einer bestehenden OpenDRIVE-Karte auf. Mit den beschriebenen Ansätzen ist eine Verbesserung bzw. Erweiterung dieser Funktionalität erreichbar. Aufgrund des modularen Aufbaus des Algorithmus A.3 („Automatisierte Extraktion von Segmenten“) ist das Ändern der bestehenden Funktionalitäten sowie das Ergänzen weiterer Funktionalitäten möglich.

### 8.2.3 Extraktion des Verhaltensattributs Speed Limit

Weiterhin bietet die in dieser Arbeit umgesetzte Methode die Möglichkeit, die Werte des Verhaltensattributs *Speed Limit* aus der OpenDRIVE-Karte zu extrahieren. Diese Extraktion erfolgt nur in bestimmten Grenzen. Es erfolgt ausschließlich eine Betrachtung des `<speed>`-Elements, das in dem zu einem BSSD-Fahrstreifen äquivalenten OpenDRIVE-Fahrstreifen definiert ist. Sofern dieses Element nicht definiert ist, wird keine Ausprägung von *Speed Limit* bestimmt. Wie sich bei der Validierung der realen Szenarien gezeigt hat, wird das `<speed>`-Element häufig nur in den Hauptfahrstreifen einer Straße definiert. Somit war auch nur in diesen Fahrstreifen eine Extraktion von *Speed Limit* möglich. Es wurden 39 % (Autobahnabschnitt, siehe Tabelle 7-2) bzw. 46 % (urbane Kreuzung, siehe Tabelle 7-4) aller Ausprägungen gefunden. Zur Verbesserung des Anteils der automatisch abgeleiteten Ausprägungen von *Speed Limit* sind zwei Erweiterungen zu der aktuell umgesetzten Methode denkbar. Einerseits ist es in dem angesprochenen Fall, dass *Speed Limit* nur in den Hauptfahrstreifen extrahiert wird, möglich, die Ausprägungen des Verhaltensattributs *Speed Limit* der restlichen BSSD-Fahrstreifen, die zur gleichen Fahrbahn wie diese Hauptfahrstreifen gehören (z.B. Randstreifen, Sperrfläche, Parkfläche), von den Hauptfahrstreifen zu übernehmen. Andererseits ist eine Bestimmung der Ausprägungen von *Speed Limit* zusätzlich anhand der in der OpenDRIVE-Karte hinterlegten Verkehrszeichen möglich.

Eine Schwierigkeit bei der Extraktion von *Speed Limit* ist die Bestimmung der Ausprägung entgegen der vorgesehenen Fahrtrichtung. Dafür ist notwendig, für jeden BSSD-Fahrstreifen zu beurteilen, ob es möglich ist, dass ein Fahrzeug von diesem BSSD-Fahrstreifen auf einen BSSD-Fahrstreifen der Gegenrichtung, sofern dieser existiert, gelangt. Bei der umgesetzten Methode erfolgt diese Beurteilung anhand von zwei Kriterien, die auf bestimmten Annahmen bzw. Vereinfachungen beruhen. Das erste Kriterium beurteilt die Trennung eines BSSD-Fahrstreifens von der Gegenrichtung anhand des in der Mitte der Straße befindlichen OpenDRIVE-Fahrstreifens. Sofern dieser nicht-befahrbar ist, wird davon ausgegangen, dass die beiden Fahrtrichtungen getrennt sind. Je nach Klassifizierung der Befahrbarkeit der OpenDRIVE-Fahrstreifen sind Anwendungsfälle denkbar, bei denen diese Annahme nicht zutrifft und ein Überfahren des nicht-befahrbaren OpenDRIVE-Fahrstreifens in der Mitte der Straße dennoch möglich ist. Das zweite Kriterium basiert auf der Klassifizierung der lateralen Begrenzungen der BSSD-Fahrstreifen als *überfahrbar* bzw. *nicht-überfahrbar*. Dabei erfolgt

---

die Bestimmung von nicht-überfahrbaren BSSD-Fahstreifenbegrenzungen hauptsächlich anhand der in der OpenDRIVE-Karte definierten Objekte. Diese Betrachtung erfolgt vereinfachend, weswegen falsche Klassifizierungen der Überfahrbarkeit möglich sind. So werden nur solche BSSD-Fahstreifenbegrenzungen als nicht-überfahrbar betrachtet, bei denen über den gesamten longitudinalen Definitionsbereich mindestens ein Objekt im Bereich der Fahstreifenbegrenzung definiert ist. Beispielsweise wird eine Fahstreifenbegrenzung, innerhalb der mehrere Pfosten in geringem longitudinalen Abstand existieren, somit nicht als nicht-überfahrbar erkannt. Darüber hinaus ist die Betrachtung der in der OpenDRIVE-Karte vorhandenen Objekte nur vereinfachend. So werden alle Objekte als Bounding-Box betrachtet, woraus bei bestimmten Objekten, wie beispielsweise bei einer Straßenlaterne, eine im Vergleich zur Realität deutlich größere Ausdehnung resultiert. Weiterhin wird nur die Ausdehnung der Objekte in Richtung der  $s$ -Koordinate betrachtet (siehe Abbildung 6-17 und Tabelle 6-3). Somit sind Fehler bei der Zuordnung von Objekten, die eine relevante Ausdehnung in Richtung der  $t$ -Koordinate besitzen, zu den BSSD-Fahstreifenbegrenzungen möglich. Diese Zuordnung erfolgt ausschließlich anhand des Abstands des Ursprungs der Objekte zu den BSSD-Fahstreifenbegrenzungen.

Insgesamt betrachtet ergibt sich, dass die automatisierte Extraktion des Verhaltensattributs *Speed Limit* in gewissen Grenzen korrekt funktioniert. Insbesondere die Ableitung von *Speed Limit* entgegen der vorgesehenen Fahrtrichtung eines BSSD-Fahstreifens weist eine hohe Komplexität auf. Aufgrund dessen erfolgt die Bestimmung bei der umgesetzten Methode nur im Rahmen von bestimmten Annahmen. Die automatisierte Extraktion von *Speed Limit* demonstriert exemplarisch die Möglichkeit des automatisierten Ableitens der Ausprägungen der Verhaltensattribute.

#### 8.2.4 Ausblick

Die in dieser Arbeit umgesetzte Methode ermöglicht eine vollständige Integration der BSSD in eine bestehende OpenDRIVE-Karte. Durch den modularen Aufbau der Implementierung ist eine Weiterentwicklung der Methode möglich. Mit den in den vorigen Abschnitten angesprochenen Maßnahmen der Verbesserung bzw. Ergänzung der umgesetzten Methode ist eine Reduktion des manuellen Aufwands für die vollständige Integration der BSSD in eine vorhandene OpenDRIVE-Karte möglich. Weiterhin erscheint es durch eine Weiterentwicklung der umgesetzten Methode möglich, in Einzelfällen eine vollständig automatisierte Durchführung der BSSD-Integration in eine OpenDRIVE-Karte zu erreichen. Dies umfasst die vollständige Segmentierung sowie Ableitung der Verhaltensattribute. Für das Erreichen dieses Zustands ist vor allem das automatisierte Ableiten der Ausprägungen der Verhaltensattribute umzusetzen. Dies wird näher bei Glatzki et al.<sup>98</sup> beschrieben.

Die Weiterentwicklung der Methode in einer Form, sodass eine vollständig automatisierte Durchführung der BSSD-Integration für alle möglichen OpenDRIVE-Karten resultiert, ist aus zwei Gründen schwer zu realisieren. Die erste Ursache ist, dass OpenDRIVE ein offener und somit für jeden verwendbarer Standard ist. Daraus ergibt sich, dass in der Praxis keine vollständig einheitliche Verwendung dieses Standards auftritt. An einigen Stellen der Modellierung von Karten mit OpenDRIVE sind

---

<sup>98</sup> Glatzki, F.; Winner, H.: Inferenz von Verhaltensattributen der verhaltenssemantischen Szeneriebeschreibung (2022).

---

individuelle Vorgehensweisen bezüglich der Verwendung des Standards zu erkennen. Als Beispiel dafür ist die Verwendung des `type`-Attributs zur Spezifizierung der Art eines OpenDRIVE-Fahstreifens zu nennen. Es hat sich gezeigt, dass einzelne vordefinierte Ausprägungen dieses Attributs in verschiedenen OpenDRIVE-Karten für unterschiedliche reale Repräsentation verwendet werden. Dies liegt vor allem an einer Modellierung der OpenDRIVE-Karten, die nicht vollständig korrekt entsprechend der offiziellen Spezifikation von OpenDRIVE erfolgt. Diese uneinheitliche Modellierung von OpenDRIVE-Karten erschwert eine vollkommen automatisierte Durchführung einer Methode zur Integration der BSSD in OpenDRIVE.

Die zweite Ursache, die eine vollkommen automatisierte Ausführung für alle möglichen OpenDRIVE-Karten erschwert, ist der minimale Informationsgehalt einer OpenDRIVE-Karte. Grundsätzlich ermöglicht OpenDRIVE eine detaillierte und informationsreiche Modellierung von Verkehrsnetzwerken. Jedoch ist die Angabe des Großteils der möglichen Informationen, die in der OpenDRIVE-Karte enthalten sind, optional. Beispielsweise die Angabe von Verkehrszeichen, Fahstreifenmarkierungen oder der zulässigen Höchstgeschwindigkeit ist optional. Daraus ergibt sich, dass es nicht für jede OpenDRIVE-Karte allgemein möglich ist, alle für die BSSD-Integration notwendigen Informationen vollständig automatisiert zu bestimmen. Es sei angemerkt, dass diese Einschränkung nur eine automatisierte Ausführung aller möglicherweise existierenden OpenDRIVE-Karten verhindert. Sofern die beschriebenen optionalen Informationen in einer OpenDRIVE-Karte vorhanden sind, ist die Optionalität dieser Informationen folglich keine Beschränkung für eine automatisierte Durchführung der BSSD-Integration für diese OpenDRIVE-Karte.

---

## 9 Fazit und Ausblick

---

In dieser Masterthesis wurden drei Konzepte abgeleitet, die eine Integration der BSSD in OpenDRIVE ermöglichen. Aus der vollständigen Umsetzung eines der drei Konzepte resultierte ein Software-Tool, das anhand automatisierter sowie manuell durchzuführender Schritte eine vollständige und korrekte Integration der BSSD in eine bestehende OpenDRIVE-Karte ermöglicht.

Dazu wurden zu Beginn übergeordnete Anforderungen sowie daraus abgeleitete Detailanforderungen an eine Integration der BSSD in ein hochgenaues Kartenformat definiert. Anhand der Detailanforderungen wurde OpenDRIVE als ein spezielles hochgenaues Kartenformat bezüglich der Erfüllung dieser Anforderungen analysiert. Es hat sich gezeigt, dass die bestehenden Strukturen von OpenDRIVE prinzipiell für eine BSSD-Integration geeignet sind, jedoch keine Möglichkeit der Integration des Verhaltensraums bieten. Aufgrund der Möglichkeit, eine OpenDRIVE-Karte durch benutzerdefinierte Strukturen zu erweitern, erschien die Realisierung aller Detailanforderungen dennoch möglich.

Basierend auf der vorangegangenen Analyse wurden drei Konzepte abgeleitet, die eine Integration der BSSD in OpenDRIVE entsprechend der zuvor definierten Detailanforderungen ermöglichen. Das erste Konzept („<lane>-Ebene“) strebt eine direkte Integration des Verhaltensraums in die Fahrstreifen von OpenDRIVE an. Das zweite Konzept („<road>-Ebene“) basiert auf dem Einfügen der BSSD-Segmente in eine Straße der OpenDRIVE-Karte. Eine zu der bestehenden OpenDRIVE-Struktur komplett getrennte Integration der BSSD wird beim dritten Konzept („<OpenDRIVE>-Ebene“) angestrebt. Für den Vergleich der drei Konzepte wurden anhand der Betrachtung der charakteristischen Merkmale der drei Konzepte sechs Bewertungskriterien definiert und bezüglich ihrer Relevanz im Kontext dieser Arbeit gewichtet. Anhand der Anwendung der gewichteten Bewertungskriterien auf die drei Konzepte ergab sich, dass das zweite Konzept („<road>-Ebene“) den höchsten Gesamtnutzen aufweist. Nach anschließender Überprüfung der Robustheit des Ergebnisses sowie der Betrachtung der inhaltlichen Ursachen dafür wurde dieses Konzept für die Umsetzung ausgewählt.

Das ausgewählte Konzept wurde anschließend als Grundlage für die Implementierung ausdetailliert. Dazu wurde eine Abfolge von auszuführenden Schritten sowie die für diese Schritte notwendigen Algorithmen für die Umsetzung des Konzepts abgeleitet. Jeder Konzept-Schritt ergab sich dabei aus dem Einfügen eines Elements der BSSD-Struktur des Konzepts in die OpenDRIVE-Karte. Die Bewertung der Umsetzbarkeit der abgeleiteten Konzept-Schritte und Algorithmen ergab die Notwendigkeit der Ergänzung des ausdetaillierten Konzepts um zwei Algorithmen zum manuellen Bearbeiten von automatisiert erzeugten Ergebnissen, um eine vollständige Umsetzung des Konzepts im Rahmen dieser Arbeit zu ermöglichen.

Basierend auf dem ausdetaillierten Konzept erfolgte dessen Implementierung als ein Software-Tool in der Programmiersprache *Python*. Das Software-Tool wurde mit einer modularen Struktur entsprechend der Struktur des ausdetaillierten Konzepts realisiert. Daraus ergab sich eine sequenzielle Abfolge von implementierten Funktionalitäten, die im Folgenden erläutert werden.

Nach dem Einlesen einer OpenDRIVE-Karte erfolgt zunächst eine Klassifizierung der OpenDRIVE-Fahrstreifen dieser Karte bezüglich der Zugehörigkeit zur Fahrbahn, um die Modellierungsgrenzen

---

der BSSD zu definieren. Da die Möglichkeit besteht, dass diese Klassifizierung fehlerhaft ist, wird die Möglichkeit des manuellen Bearbeitens der Ergebnisse der Klassifizierung der Fahrbahnzugehörigkeit gegeben. Die automatisierte Klassifizierung der Fahrbahnzugehörigkeit erfolgt anhand eines Attributs zur Beschreibung der Art eines OpenDRIVE-Fahrstreifens. Durch das Nutzen weiterer Informationen der OpenDRIVE-Karte ist eine Verbesserung dieser Funktionalität denkbar. Im nächsten Schritt erfolgt eine automatisierte Extraktion von BSSD-Segmenten, die basierend auf Regeln bezüglich des Straßenlayouts, Geschwindigkeitsbegrenzungen sowie Verkehrszeichen erfolgt. Da im Allgemeinen nicht alle BSSD-Segmente der Szenerie der OpenDRIVE-Karte automatisiert gefunden werden, folgt eine Funktionalität zum manuellen Hinzufügen und Entfernen der automatisiert gefunden BSSD-Segmente. Für eine Reduktion des Aufwands für das manuelle Erstellen von BSSD-Segmenten ist eine Erweiterung der automatisierten Extraktion von BSSD-Segmenten basierend auf zusätzlichen Informationen der OpenDRIVE-Karte möglich. In den nächsten beiden Schritten werden basierend auf den OpenDRIVE-Fahrstreifen und den zuvor erstellten BSSD-Segmenten die BSSD-Fahrstreifen definiert sowie eine eindeutige Verknüpfung zwischen diesen beiden Arten von Fahrstreifen hergestellt. Das anschließende Einfügen einer Verhaltensraum-Minimalstruktur in jeden BSSD-Fahrstreifen ermöglicht die Festlegung der Verhaltensattribute für die BSSD-relevante Szenerie der OpenDRIVE-Karte. Im letzten Schritt erfolgt die Bestimmung der Ausprägungen des Verhaltensattributs *Speed Limit* basierend auf Informationen der OpenDRIVE-Karte bezüglich der geltenden zulässigen Höchstgeschwindigkeit. Dabei ist je BSSD-Fahrstreifen sowohl die Bestimmung von *Speed Limit* entlang als auch entgegen der vorgesehenen Fahrtrichtung erforderlich. Für letzteres erfolgt eine Betrachtung der Trennung der Fahrtrichtungen einer Straße basierend auf der Klassifizierung der Fahrbahnzugehörigkeit der OpenDRIVE-Fahrstreifen sowie der in der OpenDRIVE-Karte definierten Objekte. Anschließend wird die um die BSSD-Struktur ergänzte OpenDRIVE-Karte als separate Datei abgespeichert. Für die Überprüfung der Funktionsfähigkeit der implementierten Komponenten wurden Testfälle definiert und implementiert.

Nach der vollständigen Implementierung und dem Testen des Konzepts wurde eine Verifikation und Validierung gegenüber den zu Beginn definierten Detailanforderungen und übergeordneten Anforderungen durchgeführt. Für die Verifikation wurde eine Szenerie erstellt, die einen übergeordneten Testfall als Kombination der einzelnen Testfälle repräsentiert. Durch die Verarbeitung dieser Szenerie mit dem implementierten Software-Tool war es möglich, die umgesetzte Methode gegenüber den Detailanforderungen zu verifizieren. Eine vollständige Validierung der umgesetzten Methode gegenüber den übergeordneten Anforderungen war im Rahmen dieser Arbeit nicht umsetzbar. Aufgrund dessen erfolgte eine Validierung für zwei repräsentative reale Szenarien. Einerseits wurde ein repräsentativer Autobahnabschnitt der Bundesautobahn 9 betrachtet. Dabei wurde die Validierung in zwei Stufen durchgeführt. In einer ersten Stufe wurden die manuell auszuführenden Schritte der in dieser Arbeit umgesetzten Methode übersprungen. Dabei zeigten sich einerseits Fehler bei der Klassifizierung der Fahrbahnzugehörigkeit. Andererseits zeigten sich Fehler bei der automatisierten Segmentierung, die sich aus vereinfachenden Annahmen bei der Segmentierung basierend auf Verkehrszeichen ergaben. Weiterhin wurde ein Teil der BSSD-Segmente nicht automatisiert gefunden. In einer zweiten Stufe erfolgte eine vollständige Ausführung der Methode inklusive der manuell auszuführenden Schritte. Dadurch war eine vollständige und korrekte Integration der BSSD in die OpenDRIVE-Karte

---

möglich. Die Extraktion des Verhaltensattributs *Speed Limit* war nur für Teile der Karte möglich, da die bei dieser Funktionalität verwendeten Informationen der OpenDRIVE-Karte nur für einen Teil der verwendeten Karte hinterlegt waren. Weiterhin wurde im Rahmen der Validierung eine repräsentative urbane Kreuzung der Mittelstadt Griesheim in zwei analog ausgeführten Stufen betrachtet. Die Ergebnisse waren analog zu der Validierung des Autobahnabschnitts. Zusammenfassend ergab sich eine Validierung der umgesetzten Methode in Bezug auf eine vollständige und korrekte Segmentierung für die beiden betrachteten realen Szenarien. Die Betrachtung weiterer realer repräsentativer Szenarien ist für das Anstreben des Zustands der Stichprobenvalidität nach Viehof<sup>99</sup> notwendig.

Im Rahmen einer Gesamtbewertung wurde zunächst die allgemeine Integrierbarkeit von BSSD in OpenDRIVE betrachtet. Dabei hat sich gezeigt, dass OpenDRIVE gute Grundvoraussetzungen für eine Integration der BSSD besitzt. Einerseits bietet OpenDRIVE die Möglichkeit der Erweiterung um beliebige benutzerdefinierte Strukturen. Andererseits sind die Eigenschaften des minimalen strukturellen Elements von OpenDRIVE, dem OpenDRIVE-Fahrstreifen, passend zu den Eigenschaften des minimalen strukturellen Elements der BSSD, dem atomaren Verhaltensraum. Somit war die Definition einer Struktur möglich, die mit gewissen Einschränkungen eine vollständige Integration der BSSD in eine bestehende OpenDRIVE-Karte realisiert. Im Rahmen der Verbreitung der BSSD wäre es denkbar, diese Struktur dauerhaft in den OpenDRIVE-Standard zu integrieren.

Weiterhin wurde eine Bewertung der in dieser Arbeit umgesetzten Methode vorgenommen. Dabei wurden die wichtigsten Funktionalitäten betrachtet. Die Klassifizierung der Fahrbahnzugehörigkeit erfolgte im Großteil der betrachteten Anwendungsfälle korrekt. Aufgrund von uneinheitlicher Verwendung von OpenDRIVE ergaben sich jedoch Fehler, die durch eine Erweiterung der Funktion zur Klassifizierung der Fahrbahnzugehörigkeit oder eine Ergänzung des OpenDRIVE-Standards zu vermeiden sind. Die Funktion der automatisierten Extraktion von BSSD-Segmenten funktionierte bei den betrachteten Anwendungsfällen zum Großteil korrekt. Für die Erweiterung dieser Funktionalität zur Reduktion des manuell auszuführenden Aufwands bietet sich die Verwendung weiterer Informationen der OpenDRIVE-Karte an. Insbesondere die Verwendung der in der Karte hinterlegten Fahrstreifenmarkierungen ist hierbei hervorzuheben. Die Bestimmung des Verhaltensattributs *Speed Limit* erfolgte in gewissen Grenzen korrekt, was die automatisierte Ableitung von Verhaltensattributen exemplarisch demonstrierte. Das allgemeine Vorgehen zum Ableiten von Verhaltensattributen wird bei Glatzki et al.<sup>100</sup> beschrieben.

Zusammenfassend bietet die im Rahmen dieser Arbeit umgesetzte Methode unter Beachtung bestimmter Einschränkungen die Möglichkeit der vollständigen Integration der BSSD in eine bestehende OpenDRIVE-Karte. Zur Reduktion der aktuell dabei manuell durchzuführenden Schritte ist anhand der in Abschnitt 8.2 beschriebenen Maßnahmen eine Weiterentwicklung basierend auf dem modularen Aufbau der umgesetzten Methode möglich.

---

<sup>99</sup> Viehof, M.: Diss., Objektive Qualitätsbewertung von Fahr dynamiksimulationen durch statistische Validierung (2018), S. 46.

<sup>100</sup> Glatzki, F.; Winner, H.: Inferenz von Verhaltensattributen der verhaltenssemantischen Szeneriebeschreibung (2022).

## Anhang

### A.1 Anforderungsliste

Tabelle A-1: Anforderungen an hochgenaues Kartenformat für Vollständigkeit der BSSD-Informationen

Nr.	Anforderung
(3.1)	Die Karte des Kartenformats muss basierend auf Fahrstreifen aufgebaut sein
(3.1.1)	Ein Fahrstreifen muss eindeutig identifizierbar sein
(3.1.2)	Ein Fahrstreifen muss lateral begrenzt sein
(3.1.3)	Ein Fahrstreifen muss longitudinal begrenzt sein
(3.2)	Einem Fahrstreifen müssen in lateraler Richtung ein linker benachbarter Fahrstreifen und ein rechter benachbarter Fahrstreifen zugewiesen werden können
(3.2.1)	Die longitudinalen Begrenzungen von in lateraler Richtung benachbarten Fahrstreifen müssen in longitudinaler Richtung identisch sein
(3.3)	Einem Fahrstreifen müssen in longitudinaler Richtung mehrere vorangehende und nachfolgende Fahrstreifen zugewiesen werden können
(3.4)	Es müssen beliebig viele Fahrstreifen in lateraler Richtung aneinandergereiht werden können
(3.5)	Es müssen beliebig viele Fahrstreifen in longitudinaler Richtung aneinandergereiht werden können
(3.6)	Eine laterale Begrenzung eines Fahrstreifens muss mit einer der Eigenschaften „erlaubt“, „unter Bedingungen erlaubt“, „verboten“ oder „nicht möglich“ bezüglich des Überquerens der jeweiligen Begrenzung charakterisierbar sein
(3.6.1)	Falls das Überqueren der lateralen Begrenzung mit der Eigenschaft „unter Bedingungen“ charakterisiert ist, muss eine Bedingung angegeben werden können
(3.7)	Die longitudinale Begrenzung zu Beginn eines Fahrstreifens muss mit einer der Eigenschaften „erlaubt“, „unter Bedingungen erlaubt“, „verboten“ oder „nicht möglich“ bezüglich des Überquerens dieser Begrenzung charakterisierbar sein
(3.7.1)	Falls das Überqueren der longitudinalen Begrenzung mit der Eigenschaft „unter Bedingungen“ charakterisiert ist, müssen mehrere Bedingungen angegeben werden können
(3.8)	Die longitudinale Begrenzung zu Beginn eines Fahrstreifens muss mit mehreren Eigenschaften charakterisierbar sein, die eine binäre Fallunterscheidung darstellen
(3.8.1)	Die binäre Fallunterscheidung „Lichtsignalanlage in Betrieb/nicht in Betrieb“ muss angegeben werden können
(3.8.2)	Die binäre Fallunterscheidung „Eintritt bei roter Lichtsignalanlage erlaubt/verboten“ muss angegeben werden können
(3.9)	Einem Fahrstreifen muss eine Reservierung zugeordnet werden können
(3.9.1)	Die Reservierung muss eindeutig identifizierbar sein
(3.9.2)	Die Art der Reservierung muss anhand der Ausprägungen „eigen-reserviert“, „fremd-reserviert“ und „gleich-reserviert“ angegeben werden können
(3.9.3)	Der Reservierung müssen bei einer Reservierung der Art „fremd-reserviert“ oder „gleich-reserviert“ mehrere priorisierte Verkehrsteilnehmer der Klassen

	„Kraftfahrzeug“, „Fahrradfahrer“, „Fußgänger“ und „Schienenfahrzeug“ zugeordnet werden können
(3.9.3.1)	Die möglichen Herkunfts-Fahrstreifen eines priorisierten Verkehrsteilnehmers müssen angegeben werden können
(3.9.3.2)	Die möglichen Ziel-Fahrstreifen eines priorisierten Verkehrsteilnehmers müssen angegeben werden können
(3.9.4)	Die Reservierung muss mit mehreren Eigenschaften charakterisierbar sein, die eine binäre Fallunterscheidung darstellen
(3.9.4.1)	Die binäre Fallunterscheidung „Lichtsignalanlage in Betrieb/nicht in Betrieb“ muss angegeben werden können
(3.9.4.2)	Die binäre Fallunterscheidung „Eintritt bei roter Lichtsignalanlage erlaubt/verboten“ muss angegeben werden können
(3.9.4.3)	Die binäre Fallunterscheidung „Aktive/inaktive Zusatz-Lichtsignalanlage beim Linksabbiegen“ muss angegeben werden können
(3.10)	Einem Fahrstreifen muss eine zulässige Höchstgeschwindigkeit zugewiesen werden können
(3.10.1)	Die zulässige Höchstgeschwindigkeit beim Gelten zusätzlicher Bedingungen muss angegeben werden können
(3.10.1.1)	Die zusätzliche Bedingung eines definierten Zeitintervalls muss angegeben werden können
(3.10.1.2)	Die zusätzliche Bedingung einer nassen Straße muss angegeben werden können
(3.11)	Einem Fahrstreifen muss eine Eigenschaft bezüglich der Erlaubnis eines Überholvorgangs zugewiesen werden können
(3.12)	Die Anforderungen (3.6), (3.7), (3.8), (3.9), (3.10) und (3.11) müssen für beide Fahrrichtungen innerhalb eines Fahrstreifens erfüllt sein

## A.2 Konzeptauswahl

Tabelle A-2: Paarvergleich der Bewertungskriterien, BSSD-Entwickler 1

	Zusätzliche Datenmenge	Rechenaufwand für die Positionszuordnung des BSSD-Verhaltensraums	Beeinflussung der Struktur der OpenDRIVE-Streckendefinition	Aufwand zum Einfügen der BSSD-Struktur	Aufwand zum Entfernen der BSSD-Struktur	Explizithet des BSSD-Verhaltensraums	Summe	Normiert
1	Zusätzliche Datenmenge	-	-	=	+	-	3	0,10
2	Rechenaufwand für die Positionszuordnung des BSSD-Verhaltensraums	+	+	+	+	+	10	0,33
3	Beeinflussung der Struktur der OpenDRIVE-Streckendefinition	-	-	+	+	=	7	0,23
4	Aufwand zum Einfügen der BSSD-Struktur	-	-	-	+	-	3	0,10
5	Aufwand zum Entfernen der BSSD-Struktur	-	-	-	-	-	0	0,00
6	Explizithet des BSSD-Verhaltensraums	-	=	+	+	-	7	0,23
							30	1

Tabelle A-3: Paarvergleich der Bewertungskriterien, BSSD-Entwickler 2

	Zusätzliche Datenmenge	Rechenaufwand für die Positionszuordnung des BSSD-Verhaltensraums	Beeinflussung der Struktur der OpenDRIVE-Streckendefinition	Aufwand zum Einfügen der BSSD-Struktur	Aufwand zum Entfernen der BSSD-Struktur	Explizithet des BSSD-Verhaltensraums	Summe	Normiert
1	Zusätzliche Datenmenge	-	-	-	+	=	3	0,10
2	Rechenaufwand für die Positionszuordnung des BSSD-Verhaltensraums	+	+	+	+	+	10	0,33
3	Beeinflussung der Struktur der OpenDRIVE-Streckendefinition	-	-	+	+	=	7	0,23
4	Aufwand zum Einfügen der BSSD-Struktur	-	-	-	+	=	5	0,17
5	Aufwand zum Entfernen der BSSD-Struktur	-	-	-	-	-	0	0,00
6	Explizithet des BSSD-Verhaltensraums	-	=	=	+	-	5	0,17
							30	1

Tabelle A-4: Nutzwertanalyse, Bewertung anhand Schulnotenskala

	Bewertungskriterium	Gewichtung in %	Konzept 1		Konzept 2		Konzept 3	
			<lane>-Ebene	<road>-Ebene	<road>-Ebene	<OpenDRIVE>-Ebene	<road>-Ebene	<OpenDRIVE>-Ebene
			Bewertung	Nutzwert	Bewertung	Nutzwert	Bewertung	Nutzwert
1	Zusätzliche Datenmenge	8	6	48	3	24	2	16
2	Rechenaufwand für die Positionszuordnung des BSSD-Verhaltensraums	33	6	198	4	132	2	66
3	Beeinflussung der Struktur der OpenDRIVE-Streckendefinition	21	1	21	6	126	6	126
4	Aufwand zum Einfügen der BSSD-Struktur	18	4	72	4	72	3	54
5	Aufwand zum Entfernen der BSSD-Struktur	1	1	1	5	5	6	6
6	Explizithet des BSSD-Verhaltensraums	19	2	38	4	76	5	95
	<b>Summe</b>	<b>100%</b>		<b>378</b>		<b>435</b>		<b>363</b>

### A.3 type-Attribut OpenDRIVE-Fahrstreifen

Tabelle A-5: Klassifizierung Befahrbarkeit OpenDRIVE-Fahrstreifen anhand type-Attribut

<b>type-Attribut</b>	<b>Beschreibung</b>	<b>Klassifizierung</b>
shoulder	Bankett (unbefestigt)	Nicht-befahrbar
border	Randstreifen (befestigt)	Befahrbar
driving	„normaler“ Fahrstreifen	Befahrbar
stop	Standstreifen	Befahrbar
none	Bereich ohne äquivalente reale Repräsentation	Nicht-befahrbar
restricted	Sperrfläche	Befahrbar
parking	Parkfläche/Parkplatz	Befahrbar
median	Trennstreifen	Nicht-befahrbar
biking	Fahrradweg	Befahrbar/ Nicht-Befahrbar
sidewalk	Gehweg	Nicht-befahrbar
curb	Bordstein	Nicht-befahrbar
entry	Einfädungsstreifen	Befahrbar
exit	Ausfädungsstreifen	Befahrbar
mwyEntry	Einfädungsstreifen, veraltet (Version < 1.4)	Befahrbar
mwyExit	Ausfädungsstreifen, veraltet (Version < 1.4)	Befahrbar
onRamp	Autobahnauffahrt	Befahrbar
offRamp	Autobahnabfahrt	Befahrbar
connectingRamp	Verbindungsrampe	Befahrbar
bidirectional	Zweirichtungs-Fahrstreifen	Befahrbar
roadWorks	Fahrstreifen innerhalb einer Baustelle	Befahrbar
tram	Straßenbahn	Befahrbar
rail	Schiene	Befahrbar
bus	Busfahrstreifen	Befahrbar
taxi	Taxi	Befahrbar
HOV	Fahrgemeinschaftsfahrstreifen, HOV = „High-occupancy vehicle“	Befahrbar

## A.4 Regeln automatisierte Extraktion BSSD-Segmente

Tabelle A-6: Regeln für automatisierte Extraktion von BSSD-Segmenten

	<b>Regel</b>
1	Bei dem Definitionsbeginn der ersten <i>lane section</i> einer Straße, die befahrbare OpenDRIVE-Fahstreifen beinhaltet, ist die Definition eines BSSD-Segments erforderlich
2	Bei dem Definitionsbeginn jeder <i>lane section</i> , deren Gesamtanzahl der befahrbaren OpenDRIVE-Fahstreifen sich im Vergleich zur vorangehenden <i>lane section</i> ändert, ist die Definition eines BSSD-Segments erforderlich
3	Bei dem Definitionsbeginn jeder <i>lane section</i> , die eine vorangehende <i>lane section</i> besitzt, und einen befahrbaren OpenDRIVE-Fahstreifen beinhaltet, der keinen vorangehenden OpenDRIVE-Fahstreifen besitzt (kein Verweis im Element <predecessor>), ist die Definition eines BSSD-Segments erforderlich
4	Bei dem Definitionsende jeder <i>lane section</i> , die eine nachfolgende <i>lane section</i> mit mindestens einem befahrbaren OpenDRIVE-Fahstreifen besitzt, und einen befahrbaren OpenDRIVE-Fahstreifen beinhaltet, der keinen nachfolgenden OpenDRIVE-Fahstreifen besitzt (kein Verweis im Element <successor>), ist die Definition eines BSSD-Segments erforderlich
5	Bei dem Definitionsende jeder <i>lane section</i> , die mindestens einen befahrbaren OpenDRIVE-Fahstreifen besitzt, und die eine nachfolgende <i>lane section</i> ohne befahrbaren OpenDRIVE-Fahstreifen besitzt, muss das Definitionsende eines BSSD-Segments definiert werden
6	Bei dem Definitionsbeginn jedes <speed>-Elements, das in einem befahrbaren OpenDRIVE-Fahstreifen definiert ist und einen Wechsel der in dem OpenDRIVE-Fahstreifen zulässigen Höchstgeschwindigkeit bedingt, ist die Definition eines BSSD-Segments erforderlich
7	Bei dem Gültigkeitsbeginn jedes statischen <signal>- bzw. <signalReference>-Elements, das entsprechend des VZKat ein in Deutschland gültiges Verkehrszeichen repräsentiert, ein Indikationselement darstellt und dessen Gültigkeitsbeginn bei einer s-Koordinate definiert ist, bei der mindestens ein befahrbarer OpenDRIVE-Fahstreifen existiert, wird ein BSSD-Segment definiert
8	Bei dem Gültigkeitsbeginn jedes dynamischen <signal>- bzw. <signalReference>-Elements, dessen Gültigkeitsbeginn bei einer s-Koordinate definiert ist, bei der mindestens ein befahrbarer OpenDRIVE-Fahstreifen existiert, wird ein BSSD-Segment definiert

## A.5 Spezifikation der BSSD-Integration

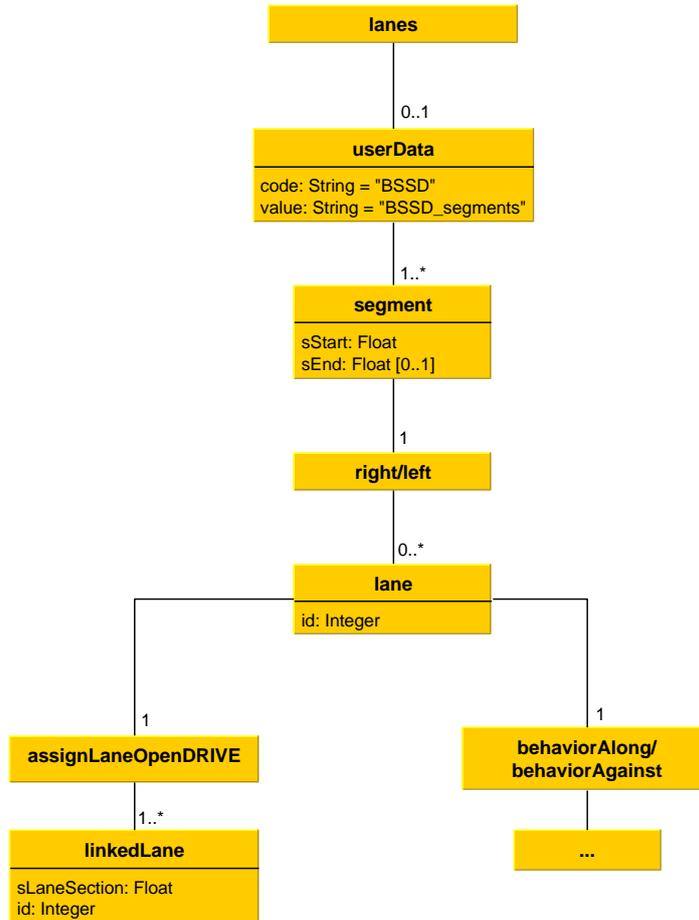


Abbildung A-1: UML-Diagramm der BSSD-Integration

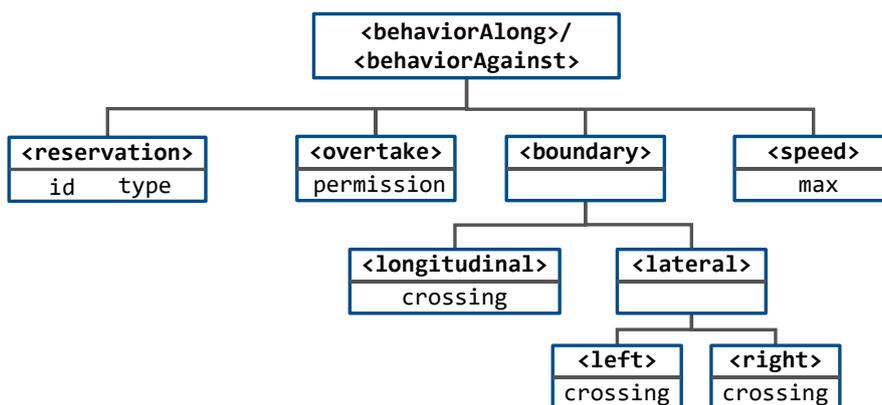


Abbildung A-2: Verhaltensraum-Minimalstruktur

Tabelle A-7: Spezifikation der Attribute der XML-Elemente der BSSD-Integration

Attribut	Datentyp	Einheit	Werte	Optional	Beschreibung
<b>&lt;userData&gt;</b>					
code	string	-	BSSD	-	Identifikation der BSSD-Integration
value	string	-	BSSD_ seg- ments	-	Identifikation der BSSD-Integration
<b>&lt;segment&gt;</b>					
sStart	float	m	[0, ∞[	-	s-Koordinate des Definitionsbeginns eines BSSD-Segments
sEnd	float	m	[0, ∞[	x	s-Koordinate des Definitionsendes eines BSSD-Segments. Verwendung, wenn auf BSSD-Segment eine BSSD-Definitionslücke folgt.
<b>&lt;lane&gt;</b>					
id	int	-	]−∞, ∞[	-	Identifikator des BSSD-Fahrstreifens. Eindeutig innerhalb des übergeordneten BSSD-Segments.
<b>&lt;linkedLane&gt;</b>					
sLane- Section	float	m	[0, ∞[	-	s-Koordinate der <i>lane section</i> , die den zu dem übergeordneten BSSD-Fahrstreifen verknüpften OpenDRIVE-Fahrstreifen beinhaltet.
id	int	-	]−∞, ∞[	-	Wert des <i>id</i> -Attributs des zu dem übergeordneten BSSD-Fahrstreifen verknüpften OpenDRIVE-Fahrstreifens.

## A.6 Gesamttestfall

Die Szenerie der Straßen 0 und 1 ist übersichtswise in Abbildung A-3 dargestellt. Der Verlauf der Straßen von unten nach oben entspricht der Betrachtungsrichtung von links nach rechts.

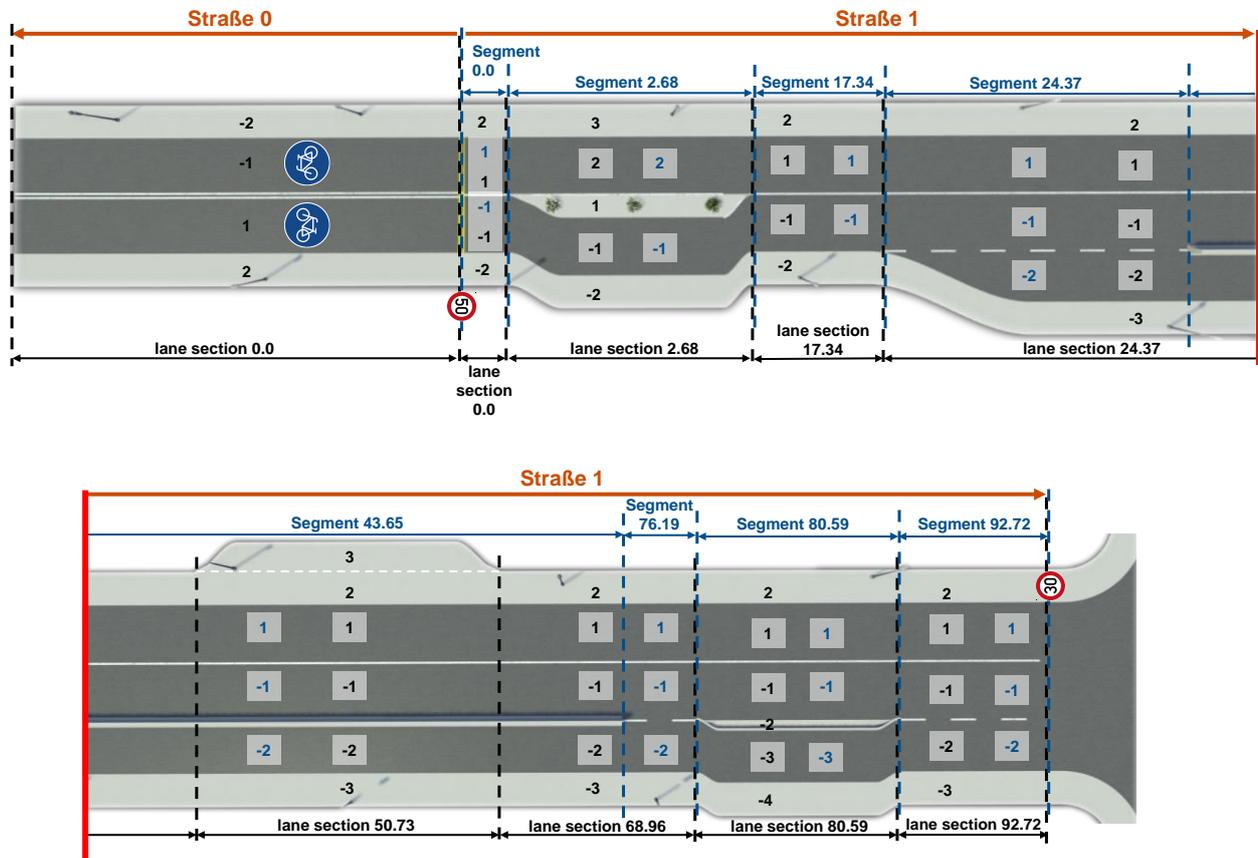


Abbildung A-3: Szenerie Straßen 0 und 1, Gesamttestfall

Die Szenerie der Straße 23 ist übersichtswise in Abbildung A-4 dargestellt. Der Verlauf der Straße von links nach rechts entspricht der Betrachtungsrichtung von links nach rechts.

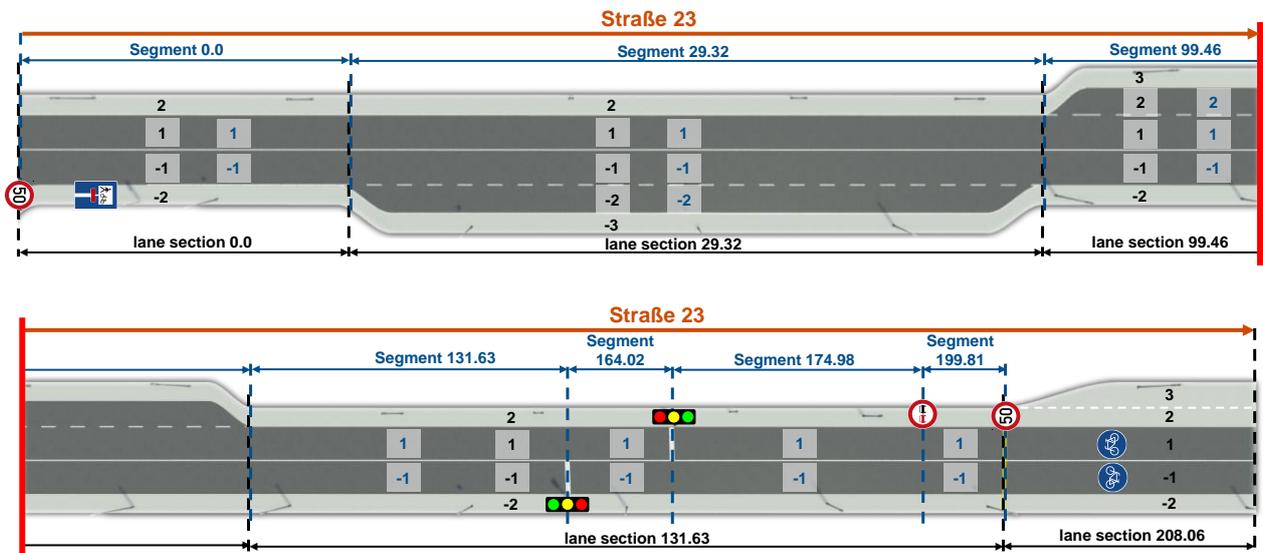


Abbildung A-4: Szenerie Straße 23, Gesamttestfall

Tabelle A-8 beinhaltet für jede Funktionalität der BSSD-Integration die zugehörige Stelle der Szenerie des Gesamttestfalls, welche diese Funktionalität testet. Die Stelle wird in Form einer *lane section* angegeben. Sofern der Test der Funktionalität auf einem bestimmten bzw. mehreren OpenDRIVE-Fahrstreifen basiert, wird das *id*-Attribut dieses bzw. werden die *id*-Attribute dieser angegeben. Wenn dem Test der Funktionalität eine definierte Stelle in der Szenerie zuweisbar ist, wird die zugehörige *s*-Koordinate angegeben.

Tabelle A-8: Zusammenhang zwischen Szenerie und zu testenden Funktionen, Gesamttestfall

Nr.	Funktionalität	Straße	lane section	id	s
1	Einlesen & Prüfen der OpenDRIVE-Datei	Gesamte Szenerie			
2	Erstellen der BSSD-Wurzelemente	Gesamte Szenerie			
		0	0.0	Alle	–
A.1	Automatisierte Suche von befahrbaren OpenDRIVE-Fahrstreifen	Gesamte Szenerie			
	Klassifizierung von Fahrradwegen	9	204.20	-2	–
A.2	Manuelles Bearbeiten der OpenDRIVE-Fahrstreifen	siehe Subfunktionen			
	Hinzufügen von befahrbaren OpenDRIVE-Fahrstreifen	Hinzufügen eines beliebigen Fahrstreifens			
	Entfernen von befahrbaren OpenDRIVE-Fahrstreifen	Entfernen dieses Fahrstreifens			
3	Erstellen der BSSD-Segmente	siehe A.3 & A.4			
A.3	Automatisierte Extraktion von Segmenten	siehe Subfunktionen			
	Segmente basierend auf der Anzahl der befahrbaren Fahrstreifen	1	0.0	Alle	–
		1	24.37	-2	–
		9	0.0	Alle	–
		9	67.50	Alle	–
		9	204.20	-2	–
		9	284.19	Alle	–
		23	0.0	Alle	–
		23	29.32	-2	–
		23	131.63	Alle	–
		23	208.06	Alle	–
	Segmente basierend auf dem Beginn/Ende eines befahrbaren OpenDRIVE-Fahrstreifens	23	29.32	-2	–
	23	99.46	2	–	
	Segmente basierend auf Geschwindigkeitsbegrenzungen	9	67.50	–	89.25
	9	204.20	–	211.83	
	Segmente basierend auf Verkehrszeichen	9	67.50	–	74.07
	23	0.0	–	9.78	
23	131.63	–	199.81		
Segmente basierend auf dynamischen Elementen des Straßenverkehrs	23	131.63	–	164.02	
23	131.63	–	174.98		
A.4	Manuelles Bearbeiten der Segmente	siehe Subfunktionen			
	Hinzufügen von Segmenten	1	2.68	–	2.68
		1	17.34	–	17.34
		1	24.37	–	43.65
		1	68.96	–	76.19
		1	80.59	–	80.59
		1	92.72	–	92.72
		9	137.76	–	137.76

		9	176.44	–	176.44
		Erstellen eines beliebigen Segments			
	Entfernen von Segmenten	Entfernen dieses Segments			
4	Erstellen der Fahrstreifengruppierungen	Gesamte Szenerie			
5	Erstellen der BSSD-Fahrstreifen	siehe A.5			
A.5	Überschneidungen zwischen <i>lane sections</i> und Segmenten finden	Gesamte Szenerie			
		9	0.0 – 27.76	Alle	–
6	Verknüpfen der OpenDRIVE- & BSSD-Fahrstreifen	siehe A.6			
A.6	Suche von zugehörigen OpenDRIVE-Fahrstreifen	Gesamte Szenerie			
		9	137.76 – 170.07	Alle	–
7	Erstellen der Verhaltensraum-Minimalstruktur	Gesamte Szenerie			
8	Befüllen des Verhaltensattributs <i>Speed Limit</i>	siehe A.7			
A.7	Automatisierte Extraktion des Verhaltensattributs <i>Speed Limit</i>	siehe Subfunktionen			
	Entlang der vorgesehenen Fahrtrichtung	Gesamte Szenerie			
	Entgegen der vorgesehenen Fahrtrichtung	Straße 1, Straße 9			
	Kriterium 1: Nicht-befahrbarer OpenDRIVE-Fahrstreifen in der Straßenmitte	1	2.68	1	–
		9	137.76	1, 2	–
	Kriterium 2: Überfahrbarkeit der BSSD-Fahrstreifenbegrenzungen	siehe Subfunktionen			
	Anhand der Nachbarfahrstreifen	1	80.59	-2	–
Anhand von Objekten	1	24.37 – 68.96	–	–	
9	Abspeichern der modifizierten OpenDRIVE-Datei	Gesamte Szenerie			



---

## Literaturverzeichnis

---

**Aeberhard, M. et al.: Experience, Results and Lessons Learned from Automated Driving on Germany's Highways (2015)**

Aeberhard, Michael; Rauch, Sebastian; Bahram, Mohammad; Tanzmeister, Georg; Thomas, Julian; Pilat, Yves; Homm, Florian; Huber, Werner; Kaempchen, Nico: Experience, Results and Lessons Learned from Automated Driving on Germany's Highways, in: IEEE Intelligent Transportation Systems Magazine (1), Jahrgang 7, S. 42–57, 2015

**Althoff, M. et al.: Automatic Conversion of Road Networks from OpenDRIVE to Lanelets (2018)**

Althoff, Matthias; Urban, Stefan; Koschi, Markus: Automatic Conversion of Road Networks from OpenDRIVE to Lanelets, in: 2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), Singapore, IEEE, 2018

**ASAM e.V.: About ASAM (2021)**

Association for Standardisation of Automation and Measuring Systems: About ASAM; <https://www.asam.net/>, 2021, Zugriff 28.10.2021

**ASAM e.V.: ASAM OpenDRIVE® (2021)**

Association for Standardisation of Automation and Measuring Systems: ASAM OpenDRIVE®; <https://www.asam.net/standards/detail/opendrive/>, 2021, Zugriff 26.10.2021

**ASAM e.V.: ASAM OpenDRIVE® - Previous Releases (2021)**

Association for Standardisation of Automation and Measuring Systems: ASAM OpenDRIVE® - Previous Releases; <https://www.asam.net/standards/detail/opendrive/older/>, 2021, Zugriff 05.01.2022

**ASAM e.V.: ASAM OpenSCENARIO® (2021)**

Association for Standardisation of Automation and Measuring Systems: ASAM OpenSCENARIO®; <https://www.asam.net/standards/detail/openscenario/>, 2021, Zugriff 28.10.2021

**ASAM e.V.: Specification ASAM OpenDRIVE, Version 1.7.0 (2021)**

Association for Standardisation of Automation and Measuring Systems: Specification ASAM OpenDRIVE, Version 1.7.0; <https://www.asam.net/index.php?eID=dumpFile&t=f&f=4422&token=e590561f3c39aa2260e5442e29e93f6693d1cccd>, 2021, Zugriff 28.10.2021

**BASt: OpenDrive-Kartendaten Testfeld A9 (2017)**

Bundesanstalt für Straßenwesen: OpenDrive-Kartendaten Testfeld A9, Mobilitäts Daten Marktplatz; <https://service.mdm-portal.de/mdm-portal-application/publDetail.do?publicationId=2594000>, 2017, Zugriff 17.03.2022

**Bender, P. et al.: Lanelets: Efficient map representation for autonomous driving (2014)**

Bender, Philipp; Ziegler, Julius; Stiller, Christoph: Lanelets: Efficient map representation for autonomous driving, in: 2014 IEEE Intelligent Vehicles Symposium Proceedings, MI, USA, IEEE, 2014

**Bengler, K. et al.: Fahrerassistenzsysteme und Automatisiertes Fahren (2021)**

Bengler, Klaus; Dietmayer, Klaus; Eckstein, Lutz; Stiller, Christoph; Winner, Hermann: Fahrerassistenzsysteme und Automatisiertes Fahren, in: Pischinger, Stefan; Seiffert, Ulrich (Hrsg.): Vieweg Handbuch Kraftfahrzeugtechnik, Springer Fachmedien Wiesbaden, Wiesbaden, 2021

**BMBF: UNICARagil (2022)**

Bundesministerium für Bildung und Forschung: UNICARagil; <https://www.unicaragil.de/de/>, 2022, Zugriff 29.03.2022

**BMJV: Straßenverkehrs-Ordnung (StVO) (2013)**

Bundesministerium der Justiz und für Verbraucherschutz: Straßenverkehrs-Ordnung (StVO); [https://www.gesetze-im-internet.de/stvo\\_2013/index.html](https://www.gesetze-im-internet.de/stvo_2013/index.html), Zugriff 17.11.2021, 2013

---

**BMVI: Gesetz zum autonomen Fahren tritt in Kraft (2021)**

Bundesministerium für Verkehr und digitale Infrastruktur: Gesetz zum autonomen Fahren tritt in Kraft; <https://www.bmvi.de/SharedDocs/DE/Artikel/DG/gesetz-zum-autonomen-fahren.html>, 2021, Zugriff 25.10.2021

**Chaplier, J. et al.: Toward a standard: RoadXML, the road network database format (2010)**

Chaplier, Julien; That, Thomas N.; Hewatt, Marcus; Gallée, Gilles: Toward a standard: RoadXML, the road network database format, in: Kemeny, Andras; Mérienne, Frédéric; Espié, Stéphane (Hrsg.): Trends in Driving Simulation Design and Experiments, Proceedings of the Driving Simulation Conference Europe 2010, Paris, 2010

**Deutscher Bundestag: Gesetz zum autonomen Fahren (2021)**

Deutscher Bundestag: Entwurf eines Gesetzes zur Änderung des Straßenverkehrsgesetzes und des Pflichtversicherungsgesetzes – Gesetz zum autonomen Fahren; <https://dserver.bundestag.de/btd/19/274/1927439.pdf>, Zugriff 25.10.2021, 2021

**Deutsches Institut für Normung e. V.: DIN ISO 8855 (2013)**

Deutsches Institut für Normung e. V.: DIN ISO 8855: Straßenfahrzeuge - Fahrzeugdynamik und Fahrverhalten - Begriffe, Beuth Verlag, Berlin, 2013

**Deutsches Institut für Normung e. V.: DIN ISO 3166-1: Codes für die Namen von Ländern und deren Untereinheiten (2020)**

Deutsches Institut für Normung e. V.: DIN ISO 3166-1: Codes für die Namen von Ländern und deren Untereinheiten – Teil 1: Codes für Ländernamen, Beuth Verlag, Berlin, 2020

**Dillmann, N.: ASAM OpenDRIVE Webinar, Part 1 (2020)**

Dillmann, Nicco: ASAM OpenDRIVE Webinar, Part 1; <https://www.asam.net/index.php?eID=dumpFile&t=f&f=3820&token=3f7d12b0573405baf8a9adac449551262d72968d>, 2020, Zugriff 02.11.2021

**Duden Online: Definition Semantik (2021)**

Duden Online: Definition Semantik; <https://www.duden.de/rechtschreibung/Semantik>, 2021, Zugriff 17.11.2021

**Dupuis, M. et al.: OpenDRIVE 2010 and Beyond -Status and future of the de facto standard (2010)**

Dupuis, Marius; Strobl, Martin; Grezlikowski, Hans: OpenDRIVE 2010 and Beyond - Status and future of the de facto standard for the description of road networks, in: Proceedings of the Driving Simulation Conference Europe, 2010

**FGSV: Begriffsbestimmungen für das Straßen- und Verkehrswesen (2020)**

Forschungsgesellschaft für Straßen- und Verkehrswesen: Begriffsbestimmungen für das Straßen- und Verkehrswesen (BBSV), FGSV Verlag GmbH, Köln, 2020

**FZD: PRORETA 5**

Fachgebiet Fahrzeugtechnik, TU Darmstadt: PRORETA 5; <https://www.proreta.tu-darmstadt.de/proreta/index.en.jsp>, Zugriff 27.10.2021

**Glatzki, F. et al.: Behavior-Semantic Scenery Description (BSSD) (2021)**

Glatzki, Felix; Lippert, Moritz; Winner, Hermann: Behavioral Attributes for a Behavior-Semantic Scenery Description (BSSD) for the Development of Automated Driving Functions, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, IEEE, 2021

**Glatzki, F.; Lippert, M.: Behavior-Semantic Scenery Description v0.4 (unveröffentlicht) (2021)**

Glatzki, Felix; Lippert, Moritz: Behavior-Semantic Scenery Description v0.4 Draft (unveröffentlicht), 2021

**Glatzki, F.; Winner, H.: Inferenz von Verhaltensattributen der verhaltenssemantischen Szenariebeschreibung (2022)**

Glatzki, Felix; Winner, Hermann: Inferenz von Verhaltensattributen der verhaltenssemantischen

---

Szeneriebeschreibung für die Entwicklung automatisierter Fahrfunktionen, akzeptiert, in: 14. Workshop Fahrerassistenzsysteme und automatisiertes Fahren, Uni-DAS e.V., Berkheim, 2022

**Google LLC: Google Maps (2022)**

Google LLC: Google Maps; <https://www.google.com/maps>, 2022, Zugriff 17.03.2022

**Hakuli, S.; Krug, M.: Virtuelle Integration (2015)**

Hakuli, Stephan; Krug, Markus: Virtuelle Integration, in: Winner, Hermann et al. (Hrsg.): Handbuch Fahrerassistenzsysteme, Springer Fachmedien Wiesbaden, Wiesbaden, 2015

**IEEE: IEEE Standard for Floating-Point Arithmetic (2019)**

The Institute of Electrical and Electronics Engineers, Inc.: IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2019, 2019

**Korsch, U.: Verkehrszeichenkatalog (VzKat) (2017)**

Korsch, Uli: Verkehrszeichenkatalog (VzKat); <http://www.vzkat.de/2017/VzKat.htm>, 2017

**Kühnapfel, J. B.: Nutzwertanalysen in Marketing und Vertrieb (2014)**

Kühnapfel, Jörg B.: Nutzwertanalysen in Marketing und Vertrieb, Springer-Gabler, Wiesbaden, 2014

**Lippert, M. et al.: Definition von Bestehens-/Versagenskriterien (2019)**

Lippert, Moritz; Klamann, Björn; Amersbach, Christian; Winner, Hermann: Definition von Bestehens-/Versagenskriterien für das partikuläre Testen von automatisierten Fahrfunktionen, in: 9. Tagung Automatisiertes Fahren, München, 2019

**Maierhofer, S. et al.: CommonRoad Scenario Designer: An Open-Source Toolbox for Map Conversion (2021)**

Maierhofer, Sebastian; Klischat, Moritz; Althoff, Matthias: CommonRoad Scenario Designer: An Open-Source Toolbox for Map Conversion and Scenario Creation for Autonomous Vehicles, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, IEEE, 2021

**Mapillary AB: Mapillary (2022)**

Mapillary AB: Mapillary; <https://www.mapillary.com/app/>, 2022, Zugriff 17.03.2022

**Meek, D. S.; Walton, D. J.: A note on finding clothoids (2004)**

Meek, D. S.; Walton, D. J.: A note on finding clothoids, in: Journal of Computational and Applied Mathematics (2), Jahrgang 170, S. 433–453, 2004

**MSC Software: VIRES übergibt OpenDRIVE® Standard an ASAM e.V. (2018)**

MSC Software: VIRES übergibt OpenDRIVE® Standard an ASAM e.V.; <https://www.mscsoftware.com/de/news/vires-uebergibt-opendriver-standard-asam-ev>, 2018, Zugriff 28.10.2021

**Pagel, S.: OpenDRIVE Viewer (2022)**

Pagel, Sebastian: OpenDRIVE Viewer; <https://odrvviewer.io/>, 2022, Zugriff 16.03.2022

**Poggenhans, F. et al.: Lanelet2: A high-definition map framework for the future of automated driving (2018)**

Poggenhans, Fabian; Pauls, Jan-Hendrik; Janosovits, Johannes; Orf, Stefan; Naumann, Maximilian; Kuhnt, Florian; Mayr, Matthias: Lanelet2: A high-definition map framework for the future of automated driving, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, IEEE, 2018

**Pöschl, S.: Straßenquerschnitt (2008)**

Pöschl, Sascha: Straßenquerschnitt; <https://de.wikipedia.org/wiki/Stra%C3%9Fenquerschnitt>, 2008, Zugriff 06.01.2022

**Python Software Foundation: xml.etree.ElementTree — The ElementTree XML API (2022)**

Python Software Foundation: xml.etree.ElementTree — The ElementTree XML API; <https://docs.python.org/3/library/xml.etree.elementtree.html>, 2022

---

**SAE International: SAE International Standard J3016 (2014)**

SAE International: SAE International Standard J3016: Taxonomy and Definitions for Terms related to On-Road Motor Vehicle Automated Driving Systems, 2014

**Shalev-Shwartz, S. et al.: On a Formal Model of Safe and Scalable Self-driving Cars (2017)**

Shalev-Shwartz, Shai; Shammah, Shaked; Shashua, Amnon: On a Formal Model of Safe and Scalable Self-driving Cars, 2017

**Tas, O. S. et al.: Making Bertha Cooperate—Team AnnieWAY’s Entry to the 2016 Grand Cooperative Driving Challenge (2018)**

Tas, Omer S.; Salscheider, Niels O.; Poggenhans, Fabian; Wirges, Sascha; Bandera, Claudio; Zofka, Marc R.; Strauss, Tobias; Zollner, J. M.; Stiller, Christoph: Making Bertha Cooperate—Team AnnieWAY’s Entry to the 2016 Grand Cooperative Driving Challenge, in: IEEE Transactions on Intelligent Transportation Systems (4), Jahrgang 19, S. 1262–1276, 2018

**Ulbrich, S. et al.: Definition der Begriffe Szene, Situation und Szenario für das automatisierte Fahren (2015)**

Ulbrich, Simon; Menzel, Till; Reschka, Andreas; Schuldt, Fabian; Maurer, Markus: Definition der Begriffe Szene, Situation und Szenario für das automatisierte Fahren, in: 10. Workshop Fahrerassistenzsysteme FAS 2015, Walting, 2015

**Viehof, M.: Diss., Objektive Qualitätsbewertung von Fahrdynamiksimulationen durch statistische Validierung (2018)**

Viehof, Michael: Objektive Qualitätsbewertung von Fahrdynamiksimulationen durch statistische Validierung, Dissertation TU Darmstadt, Darmstadt, 2018

**Wille, J. M. et al.: Stadtpilot: Driving autonomously on Braunschweig's inner ring road (2010)**

Wille, Jorn M.; Saust, Falko; Maurer, Markus: Stadtpilot: Driving autonomously on Braunschweig's inner ring road, in: 2010 IEEE Intelligent Vehicles Symposium, La Jolla, CA, USA, IEEE, 2010

**Ziegler, J. et al.: Making Bertha Drive — An Autonomous Journey on a Historic Route (2014)**

Ziegler, Julius; Bender, Philipp; Schreiber, Markus; Lategahn, Henning; Strauss, Tobias; Stiller, Christoph; Dang, Thao; Franke, Uwe; Appenrodt, Nils; Keller, Christoph G.; Kaus, Eberhard; Herrtwich, Ralf G.; Rabe, Clemens; Pfeiffer, David; Lindner, Frank; Stein, Fridtjof; Erbs, Friedrich; Enzweiler, Markus; Knoppel, Carsten; Hipp, Jochen; Haueis, Martin; Trepte, Maximilian; Brenk, Carsten; Tamke, Andreas; Ghanaat, Mohammad; Braun, Markus; Joos, Armin; Fritz, Hans; Mock, Horst; Hein, Martin; Zeeb, Eberhard: Making Bertha Drive — An Autonomous Journey on a Historic Route, in: IEEE Intelligent Transportation Systems Magazine (2), Jahrgang 6, S. 8–20, 2014