

KoMoD Abschlussbericht

01.07.2017. – 30.06.2019

Zuwendungsempfänger: DLR

Partner DLR

Förderkennzeichen: **16 AVF 1006 H**



Bundesministerium
für Verkehr und
digitale Infrastruktur

Autor: Henning Mosebach, Jan Lauermann, David
Suske, Fabian de Ponte Müller
Version: 1.0
Stand: 20.01.2020



Dokumenteninformation

Dokument	DLR Abschlussbericht KoMoD		
Dateiname	DLR_AU_TS_Abschlussbericht_KoMoD_V01.docx		
Autor(en)	Henning Mosebach, Andreas Richter, Michael Scholz, Jan Lauer- mann, Qinrui Tang, David Suske, Fabian de Ponte Müller		
Zustand	Final	Version	01.00.00
Erstellt am	15.10.2019	Letzte Änderung am	22.01.2020

Änderungsübersicht

Nr.	Datum	Version	Änderungsgrund	Durchgeführt
1	15.10.2019	0.1	Erstellung und Gliederung	Mosebach
2	16.12.2019	0.2	Zusammenfügung der Einzelbeiträge	Lauermann, Mosebach
3	20.12.2019	0.3	Konsolidierte Zwischenfassung	Mosebach
4	22.01.2020	1.0	Fertigstellung des Abschlussberichts	Mosebach

Inhalt

1 Wichtige wissenschaftlich-technische Ergebnisse und andere wesentlicher Ereignisse	1
1.1 Fahrzeug-Kommunikationsarchitektur	1
1.2 Fahrzeugseitige Referenzimplementierung	5
1.3 Fahrzeugseitiger Routingclient.....	10
1.4 Digitale Karte (HD-Map).....	19

1 Wichtige wissenschaftlich-technische Ergebnisse und andere wesentlicher Ereignisse

1.1 Fahrzeug-Kommunikationsarchitektur

Ein Ziel des Projektes KoMoD bestand in der Nutzbarmachung von infrastrukturseitig vorhandenen und aufbereiteten Informationen des Testfeldes für automatische Fahrmanöver in AVF-Fahrzeugen (AVF=Automatisierte und Vernetzte Fahrzeuge).

Das DLR verfügt mit dem FASCarE über ein Forschungsfahrzeug welches zum Projektstart bereits Elemente zum vernetzten und automatisierten Fahren (AVF) enthielt. Durch das Einbringen von Zusatzinformationen über Kommunikation wie zum Beispiel die Daten der Ampeln, Daten über Sonderereignisse oder auch strategischen Informationen zum Routing kann der Einsatzbereich ausgebaut werden, so kann zum Beispiel der angekündigte Umschaltzeitpunkt der Ampel auf die automatisch eingestellte Fahrzeuggeschwindigkeit geschaltet werden.

Zu diesem Zwecke wurde die Kommunikationsarchitektur des DLR-Demonstrationsfahrzeugs DLR FASCarE durch Software- und Hardwareanpassungen so an das Datenangebot des Testfeldes Düsseldorf angepasst, dass sowohl über den Kommunikationsstandard ITS G5 als auch über Mobilfunk entsprechend aufbereitete Daten empfangen wurden. Nötig waren dafür Anpassungen der Empfangsmodule und entsprechende Softwareänderungen.

Zum Projektstart von KoMoD wurden die funktionellen Anforderungen an die Kommunikation zwischen der Infrastruktur des Testfeldes Düsseldorf und dem DLR Forschungsfahrzeug FASCarE aufgestellt und analysiert und mit den Betreibern der Datenquellen- und Senken einheitliche und skalierbare Schnittstellen festgelegt.

Diese funktionellen Anforderungen bestanden in der Versorgung des Fahrzeugs mit strategischen Informationen von zentralen Verkehrsrechnern über Mobilfunk sowie in dem angestrebten Empfang taktisch/dynamischer Informationen über ITS G5 ad-hoc Automotive Netze.

Für die Mobilfunkverbindungen von und zur Infrastruktur waren sowohl Verkehrsdatenplattformen vorgesehen (MDM-Server, Verkehrsmanagement- und Telematik-Server) als auch Sonderrechner wie z.B. ein Geodatenserver oder ein spezieller Server eines Projektpartners „HAV-Schicht“ genannt, der exemplarisch als Provider für aufbereitete und dynamische Daten agiert. Zu diesen Servern werden Mobilfunkverbindungen projektiert und die entsprechenden Datenaustauschverfahren spezifiziert.

Zusätzlich wurde die Versorgung des Fahrzeugs mit taktischen Informationen über ITS G5 projektiert. Dazu wurden LSA des Testfeldes Düsseldorf ausgewählt die die ETSI-Standard Nachrichten SPaT und MAP versenden. Es wurden die folgenden Standards definiert:

- CAM: EN 302 637-2 v1.3.2 (2014-11)
- DENM: EN 302 637-3 v1.2.2 (2014-11)
- SPAT: ETSI TS 103 301 V1.1.1, (2016-11), ISO/CEN 19091 (2017-03)
- MAP: ETSI TS 103 301 V1.1.1, (2016-11), ISO/CEN 19091 (2017-03)

- IVI: ETSI TS 103 301 V1.1.1, (2016-11), ISO TS 19321 (2015-08). (encoding mittels EN ISO 14823:2017-08).

Über diese Standards hinaus wurden fahrzeugseitige Zusatzinformationen, die vom Wert für das lokale verkehrliche Mikromanagement sein können, in die ETSI Nachrichten hineinkodiert und speziell in CAM oder DENM zur Erweiterung der verkehrlichen Erfassung projektiert. Dies sind z.B. eine Stauendeerkennung in DENM, der der aktuelle Automationsgrad des Fahrzeuges oder der Qualitätsgrad des gerade befahrenen Straßenabschnitts in CAM oder DENMS.

Zusammenfassung der technischen Anforderungen an die fahrzeugseitige Kommunikationsplattform.

- Die Kommunikationsplattform muss für automotiveübliche Bedingungen im Sinne Vibrationsfestigkeit und Temperaturbeständigkeit ausgelegt sein. Die Servicebarkeit muss dadurch gewährleistet sein, dass alle Anschlüsse zugänglich sind und dass einfache Konfigurier- und Loggingdaten über ein Browserähnliches Interface eingespielt oder monitort werden können
- Das Entwicklungsframework der Kommunikationsplattform muss es ermöglichen, dass in den unterschiedlichen Protokollschichten auch während der Pilotphase des Projektes noch Anpassungen durch neu auftretende Datenquellen oder Optimierungsmöglichkeiten vollzogen werden können
- Es muss eine Applikationsschicht vorgesehen sein, auf der logische Verknüpfungen und Loggingfunktionalitäten erstellt werden können, ohne dass lizenzrechtliche Verhandlungen geführt werden müssen. Dadurch kann z.B. flexibel die Analyse der Qualität der unterschiedlichen Schaltzeitprognosen durchgeführt werden.

Eine Grobübersicht über die Kommunikationspfade für die taktisch/dynamischen Daten von der Infrastruktur zur Fahrzeugautomation ist in Abbildung 1 dargestellt.

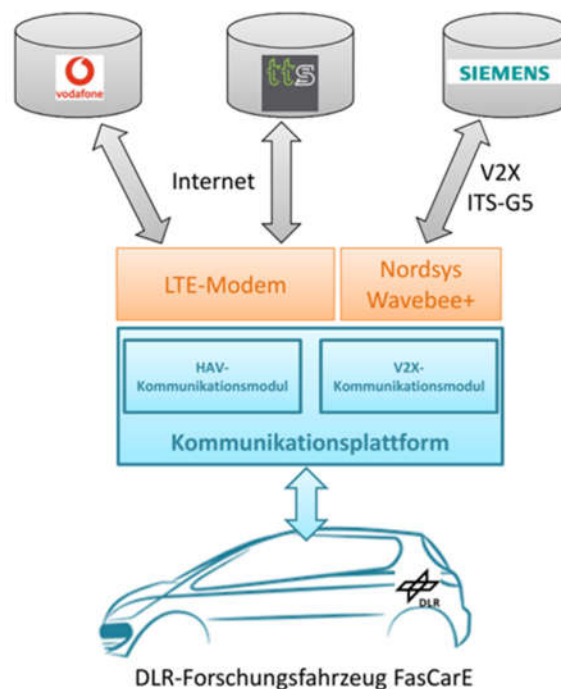


Abbildung 1: Kommunikationskanäle für die taktisch/dynamischen zwischen Testfeld Düsseldorf und DLR-Fahrzeug mit HAV-Modul und V2X-Modul

Nach den spezifizierenden Arbeiten wurden die Entwicklung und Integration des V2X-Kommunikationsmoduls und des HAV-Kommunikationsmoduls durchgeführt und abgeschlossen. Des Weiterem wurde eine Visualisierung der zentralbasierte- und lokalen LSA Schaltzeitprognose entwickelt. Die Arbeiten werden in den nächsten drei Abschnitten detaillierter erläutert.

Das V2X-Kommunikationsmodul ist für die Anbindung an die Wavebee+ V2X-Einheit zuständig. Dabei nutzt die Software die Bibliotheken aus einem V2X Stack von Projektpartnern.

Die Kommunikation zwischen dem V2X-Kommunikationsmodul und dem FasCarE Forschungsfahrzeug läuft über einer UDP/IP Schnittstelle. In der Payload der UDP Pakete werden die für die Automatisierung benötigten Daten aus der MAPEM und SPATEM Nachrichten mittels einer „KOMOD_ToDominion“ C-Struktur in das DLR Entwicklungsframework eingegeben. Zu den SPATEM Informationen zählen die Kreuzungs-ID, die aktuelle Ampelphase und die geschätzte Schaltzeit für jedes Signal, sowie die distanzabhängige Geschwindigkeitsempfehlung. Zu den übermittelten MAPEM Informationen zählen die Kreuzungs-ID, die geographischen Referenzkoordinaten, die Fahrbahnpolygone, die erlaubten Fahrmanöver und die Verbindungen zwischen eingehende und ausgehende Fahrbahnen. Die komplette C-Struktur Definition kann dem Anhang XX entnommen werden. Die UDP Pakete mit den SPATEM/MAPEM Informationen werden an das FasCarE an zuvor definierte Ports 52100 gesendet.

Das V2X-Kommunikationsmodul empfängt auch Informationen zur Lokalisierung aus dem FasCarE. Die Informationen werden mittels einer „KOMOD_ToComm“ C-Struktur in das DLR Entwicklungsframework in eine UDP Payload codiert und an UDP-Port 52000 gesendet. Zu den Informationen zählen die geographischen Koordinaten und deren Konfidenzen, sowie die Geschwindigkeit, Bewegungsrichtung, Drehrate, Beschleunigungen und deren Konfidenzen.

Die V2X-Kommunikationsmodul-FasCarE Schnittstelle wurde vor den jeweils Feldtests im Labor anhand von bereits vorhandenen RSUs getestet.

Das HAV-Kommunikationsmodul ist für die Anbindung an die HAV-Schicht und somit zum Empfang von zentral berechneten Schaltzeitprognosen zuständig. Bei der Entwicklung wurden die unter beschriebene Schnittstellen mit SPATEM-UDP Nachrichten in C# programmiert. Da die Kreuzungstopologiedaten auf der HAV-Schicht nicht dem ETSI SPAT Format entsprachen, wurden diese nicht decodiert. Dadurch bekommt das FASCarE die Informationen zu der Kreuzungstopologie ausschließlich über das V2X-Kommunikationsmodul. Das HAV-Kommunikationsmodul wurde auf ein PICE105 Windows-Industrie PC mit LTE-Anbindung portiert.

Auch die HAV-Kommunikationsmodul-FasCarE Schnittstelle wurde vor den Feldtests im Labor getestet, dabei wurde bereits auf die Mobilfunkschnittstelle des Testfeldes Düsseldorf zugegriffen.

Zur einfachen und schnellen Verifizierung der lokalen und zentralbasierte Ampelphasenprognosen bei den Feldtests wurde eine dedizierte Visualisierung der LSA Daten (Ist-Phasen und Schaltzeitprognose) in JAVA programmiert. Diese empfängt UDP Pakete mit der aktuellen Phase und die Zeit zum nächsten Phasensprung von den HAV- und V2X-Kommunikationsmodulen und zeigt Diese auf einer graphischen Oberfläche auf einem Bildschirm an. Da das V2X-Kommunikationsmodul alle empfangene SPATEM Nachrichten ungefiltert weiterleitet, wird in der Schaltzeitvisualisierung nur die relevante Signalgruppe in Abhängigkeit der aktuellen Position und Fahrtrichtung und unter Berücksichtigung der vorgegebenen Route angezeigt.

Durch die Analyse und Visualisierung der eingegangenen Ampel Ist-Daten sowie dessen Schaltzeitprognosen mit den infrastrukturbetreibenden Projektpartnern konnte so eine Verbesserung der Prognosequalität durch die Infrastrukturbetreiber erzielt werden konnte.

Das V2X-Kommunikationsmodul ist für die Anbindung an die Wavebee+ V2X-Einheit zuständig. Dabei nutzt die Software die Bibliotheken aus dem Nordsys V2X Stack. Das Modul muss folgende Aufgaben erledigen:

- Empfangen von SPAT und MAP Nachrichten von den RSU an den LSAs und weiterleiten der Information an das DLR-Fahrzeug
- Weiterleiten von DENM Payloads aus dem DLR-Fahrzeug an die Wavebee+
- Weiterleiten der präzisen Ego-Position des DLR-Fahrzeugs, welche mittels Sensorfusion berechnet wurde, an die Wavebee+.
- StationID (0x00150001) und Fahrzeugtyp werden in der Wavebee+ beim Starten konfiguriert. Die CAM Nachrichten werden automatisch von der Wavebee+ Einheit generiert.
- Weiterleiten der genauen Ego-Position vom DLR-Fahrzeug an die Wavebee+ Einheit.
- Senden der aktuellen Ego-Position des Fahrzeugs an das HAV-Kommunikationsmodul

Zum Testen der V2X Kommunikation im Labor wurde eine MK5 V2X-Einheit von Cohda Wireless eingesetzt. Diese Einheit kann so konfiguriert werden, dass sie sowohl CAMs als auch SPAT, MAP und DENM Nachrichten sendet. Sie ermöglicht es auch empfangene Nachrichten in eine pcap Datei zu speichern. Diese pcap streams wurden genutzt um mit den Projektpartnern eine Analyse des Datenverkehrs für Debuggingzwecke vorzunehmen.

1.2 Fahrzeugseitige Referenzimplementierung

Neben der im vorangegangenen Kapitel beschriebenen Entwicklung der KoMoD Kommunikationsplattform im DLR Forschungsfahrzeug FASCarE wurde das fahrzeugseitig bereits vorhandene Automationssystem auf das Karree Friedrichstadt angepasst mit dem Ziel, den definierten Zielerwartungsfall, im Testfeld Düsseldorf in automatischer Längsführung zu befahren. Dabei waren für die Arbeiten auf dem Fahrzeug die folgenden Grundfunktionen entwicklungsleitend:

- Automatisches Bremsen vor statischen und dynamischen Objekten und Abstände halten
- Automatisches Fahren nach HD-Map
- Automatisches Spurhalten
- Automatische Reaktion auf Ampelphasen über I2V.
- Automatisches Linksabbiegen durch Querverkehr
- Automatischer Spurwechsel

Diese Funktionen wurden zunächst in Braunschweig und auf dem Sonderlandeplatz Peine-Eddesse mittels virtuell von Düsseldorf nach Braunschweig verlagerten Infrastrukturelementen wie z.B. HD-Map, LSA-Aufbauten und statischen Objekte entwickelt. Im weiteren Projektverlauf wurde dann die jeweilige Teilfunktion in mehreren einwöchigen Kampagnen in Düsseldorf am Karree getestet und optimiert.

Das DLR verfügt mit der Versuchsträger-Großanlage FASCar über F&E-Fahrzeuge, welche speziell für das Entwickeln und Testen von automatisierten Fahrfunktionen in vernetzten Umgebungen im öffentlichen Raum konzipiert wurden. Die sensorische Ausstattung des im Projekt KoMoD verwendeten Fahrzeugs FASCarE ist in Abbildung 2 dargestellt.

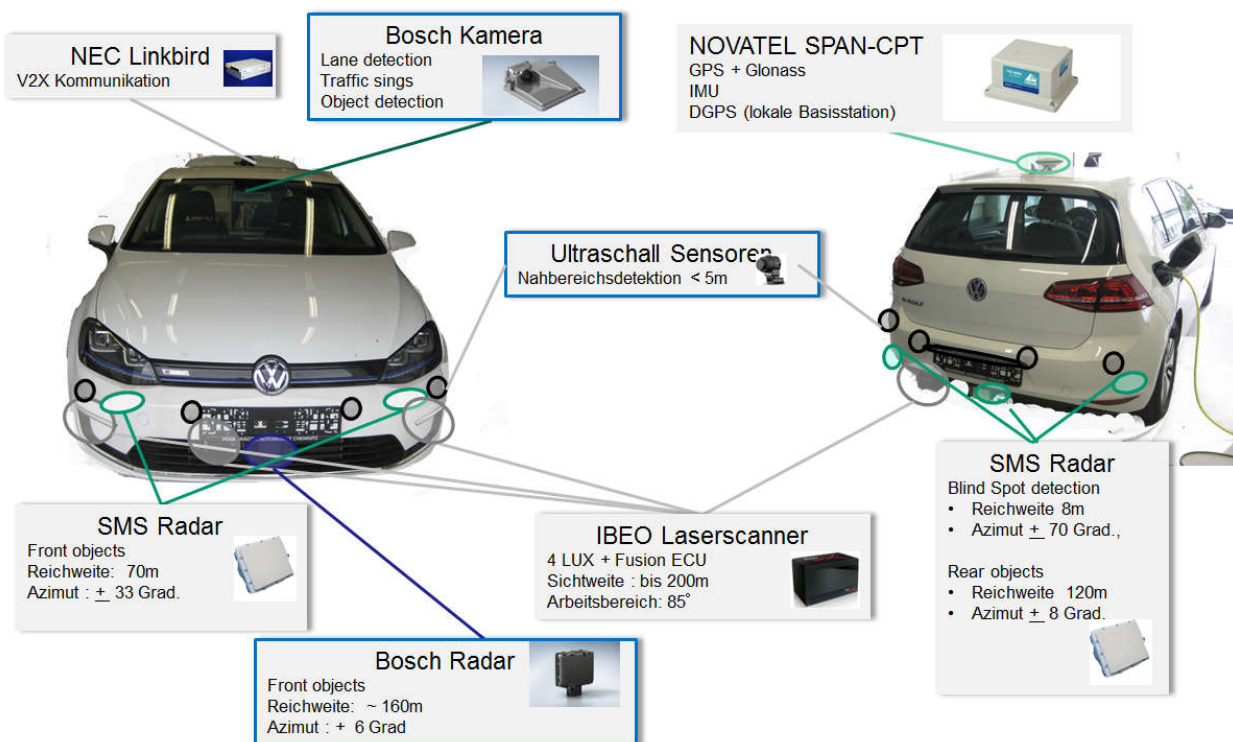


Abbildung 2: Sensorausstattung des FASCarE

Wie bereits im vorherigen Kapitel beschrieben verfügt das Fahrzeug, neben den dargestellten Sensoren, zur Erweiterung des Wahrnehmungshorizontes der Fahrzeugautomatisierung über eine Kommunikationsplattform, welche mit Mobilfunk und ITS G5 Schnittstellen ausgestattet ist. Hierüber kann durch eine gezielte Versorgung mit lokalen Informationen an Stellen besonderer Relevanz eine Erhöhung der Informationslage im Fahrzeug erzielt werden, wodurch wichtige Eingangsdaten für die automatisierte Fahrfunktion gegeben werden können.

Diese umfangreiche Sensor- und Kommunikationsausstattung ist die Ausgangsbasis für die Umsetzung komplexer automatischer Fahrfunktionen im urbanen Raum. Um diese Aufgabe zu erfüllen erfolgen zahlreiche Informationsverarbeitungsschritte. So werden beispielsweise die gesammelten Sensordaten und gespeicherte Offline-Daten zu einer Umgebungsrepräsentation zusammengeführt, Prädiktionen errechnet, Navigations- und Bewegungsplanung durchgeführt, Regelungsstellgrößen generiert, Notfallmanöver geplant und Systemüberwachung durchgeführt. All diese Bearbeitungsschritte sind in ihrer Ausprägung und Umsetzung außerdem abhängig vom Einsatzzweck und dessen Umgebungsbedingungen. Um dieser hohen Komplexität und den ständig wandelenden bzw. wachsenden Anforderungen an eine Fahrzeugautomation Rechnung zu tragen wurde vom DLR die Softwarearchitektur zur Umsetzung der Fahrfunktionen grundlegend überarbeitet. Im Projekt KoMoD wurde dieses Konzept das erste Mal in einer komplexen urbanen Umgebung erfolgreich umgesetzt und erprobt.

Grundlegend sieht das verwendete Architekturkonzept vor die Automationssoftware überwiegend in Bibliotheken auszulagern und gegenüber der Middleware zu kapseln. Dadurch entstehen leichtgewichtige Anwendungen und ein hohes Maß an Flexibilität. Auf diese Weise lassen sich bestehende Funktionen leicht erweitern oder austauschen. Neue Funktionen können separat entwickelt und getestet sowie nach Erreichen eines einsatzfähigen Standes in das Repertoire der Kernfunktionen übernommen werden. Die Struktur der Bibliothek für automatisches Fahren zeigt Abbildung 3. Die Hauptpakete beinhalten automatische Fahrfunktionen (FUN) und ein Datenmodell für die Umgebungsrepräsentation (ENV). FUN enthält Algorithmen zur Trajektorienplanung und Trajektorienfolgeregelung, Notfallplaner sowie Navigationsplanung. ENV wiederum beinhaltet Datenmodelle für den Eigenzustand, die Straßengeometrie, Verkehrszeichen sowie Verkehrsteilnehmer.

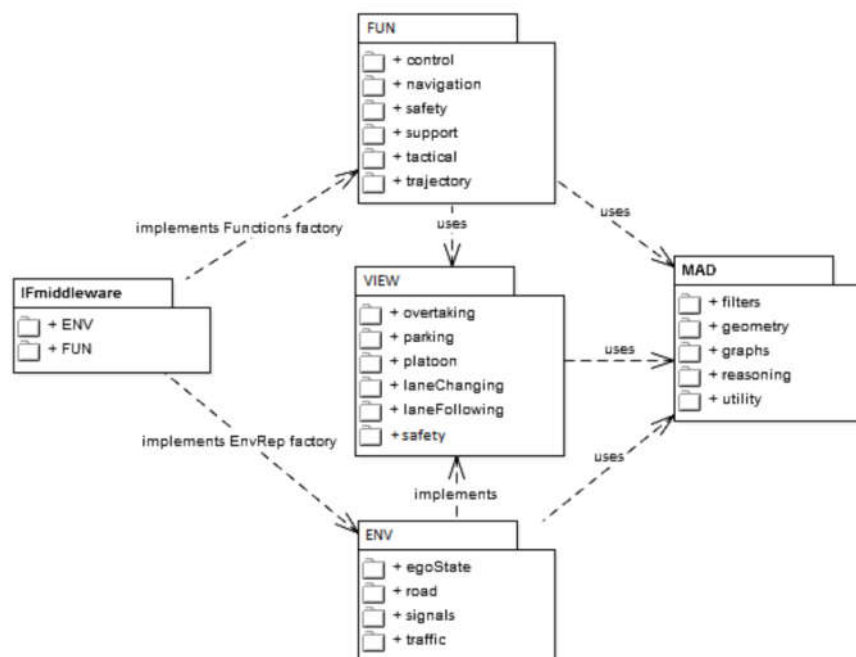


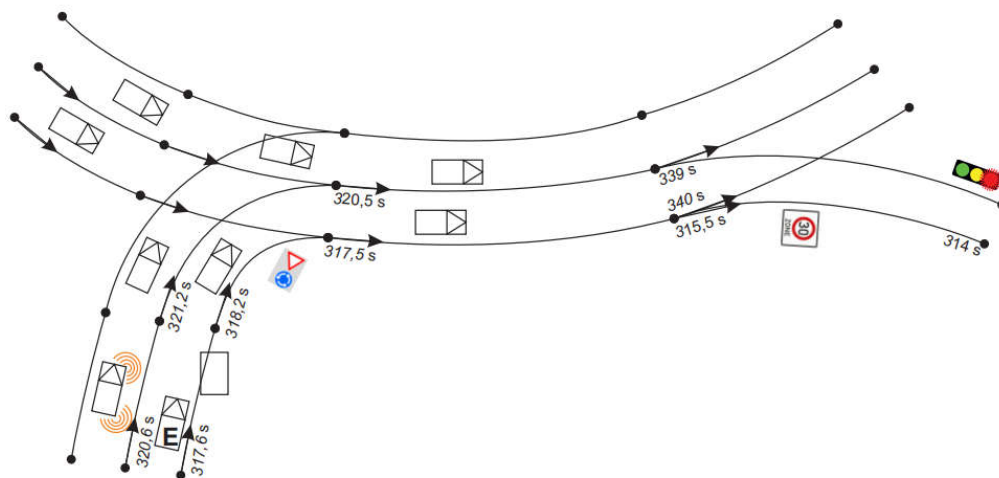
Abbildung 3: Bibliotheken des angewendeten Softwarekonzepts

Erfahrungen aus vorherigen Softwaredesigns haben gezeigt, dass eine Entkopplung zwischen Fahrfunktionen und Datenmodellen entscheidend ist. Andernfalls führt eine strukturelle Anpassung der Datenmodelle zwangsläufig zu der Notwendigkeit ebenfalls die Fahrfunktionen zu überarbeiten was in einem langwierigen fehleranfälligen Prozess endet. Die Entkopplung der Fahrfunktionen in FUN von der Umgebungsrepräsentation in ENV erfolgt über die VIEW Bibliothek. VIEW definiert verschiedene Perspektiven wie die Fahrzeugumgebung verstanden werden kann und welche Teile relevant für das geplante Fahrmanöver sind.

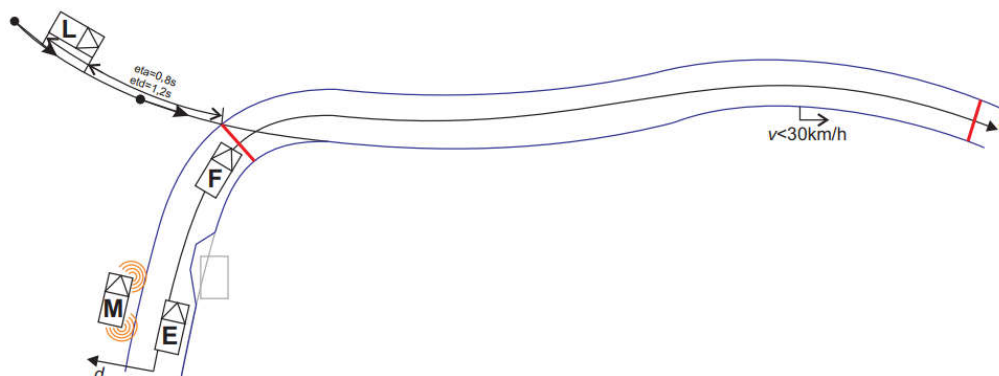
So wurde für das Projekt KoMoD der „Lane Following View“ für die Fahrt im urbanen Raum mit komplexer Verkehrssituation und Zusatzinformation der Infrastruktur weiterentwickelt. Über diese Schnittstelle werden aus der Gesamtheit aller Sensordaten und Vorabinformation die relevanten Daten herausgefiltert und für das Planen von Spurfolgemanövern an die Fahrfunktionen übergeben. Die gesamten Spurkanten und Stoplinien in der lokalen Umgebung, alle detektierten Verkehrsteilnehmer und ihre Zustände, Zustand des Egofahrzeugs, die Navigationsinformationen, alle statischen Objekte sowie alle Verkehrszeichen und Verkehrssignale werden berücksichtigt.

Abbildung 4(a) zeigt beispielhaft die ungefilterte Datenbasis im Fall der Einfahrt auf eine mehrspurige Fahrbahn. Der „Lane Following View“ identifiziert die generalisierte Spur, die Konfliktzonen auf der Spur, die zulässige Höchstgeschwindigkeit sowie die relevanten Verkehrsteilnehmer.

Abbildung 4(b) zeigt die Abstraktion der gesamten Datenbasis der aktuellen Umgebung auf die relevanten Informationen für das Spurfolgemanöver.



(a) Repräsentation der Umgebung aus dem gesamten Datenbestand



(b) Funktionsspezifische Reduzierung der Informationsmenge: Spurfolgen

Abbildung 4: Beispiel eines „Lane Following View“

Die Fahrfunktionen wurden sukzessive in Braunschweig auf dem DLR-Gelände entwickelt (siehe Abbildung 5) dann auf dem Sonderlandeplatz Peine-Eddesse um die per HD-Karte eingeblendeten Infrastrukturmerkmale ergänzt und nachfolgend in Düsseldorf getestet und optimiert.



Abbildung 5: FASCarE auf dem DLR Testgelände in Interaktion mit einer mobilen LSA

Abbildung 6 zeigt das Geschwindigkeitsprofil eines Testszenarios. Bei (a) startet das Fahrzeug die Fahrt und damit die Beschleunigung auf die Wunschgeschwindigkeit. Bei (b) ist ein Geschwindigkeitslimit erreicht, welches aus Sicherheitsgründe implementiert wurde. Bei (c) wurde eine Geschwindigkeitsempfehlung der LSA via SPaT empfangen, welche in einer Anpassung der aktuellen Geschwindigkeit resultiert. Diese wurde am Punkt (d) erreicht. Nach Passieren der LSA endet kurze Zeit später die Teststrecke. Aus diesem Grund muss die Geschwindigkeit erneut reduziert werden (e). Das Fahrzeug kommt entsprechend am Ende der Teststrecke zum Stillstand (f).

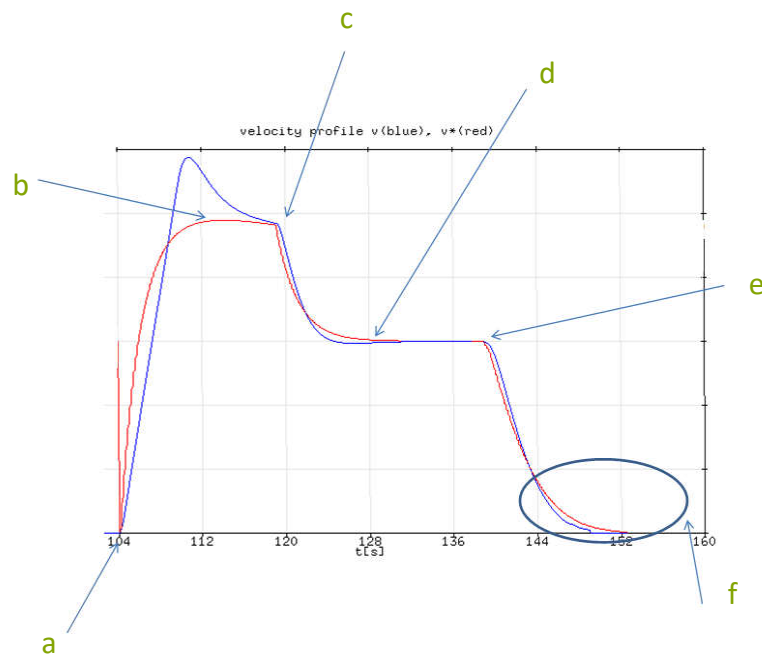


Abbildung 6: Geschwindigkeitsprofil bei Fahrt auf DLR Prüfgelände und Empfangen von SPaT-Nachrichten, geplante Geschwindigkeit (blau), aktuelle Geschwindigkeit (rot)

Das überarbeitete Softwarekonzept mit der funktionspezifischen Reduzierung der Sensordatenmenge wurde zusammen mit den vorabgetesteten Automationsfunktionen auf dem Versuchsfahrzeug FASCarE implementiert und auf dem Testfeld Düsseldorf erfolgreich erprobt. Abbildung 7 zeigt den „Lane Following View“ in einem ebenfalls für das Projekt KoMoD entwickelten Live-Birdeye-Viewer auf dem Testfeld Düsseldorf. Zu erkennen ist, dass unter anderem der LSA-Status via C2X-Kommunikation in der Bahnfolgeplanung berücksichtigt wird.

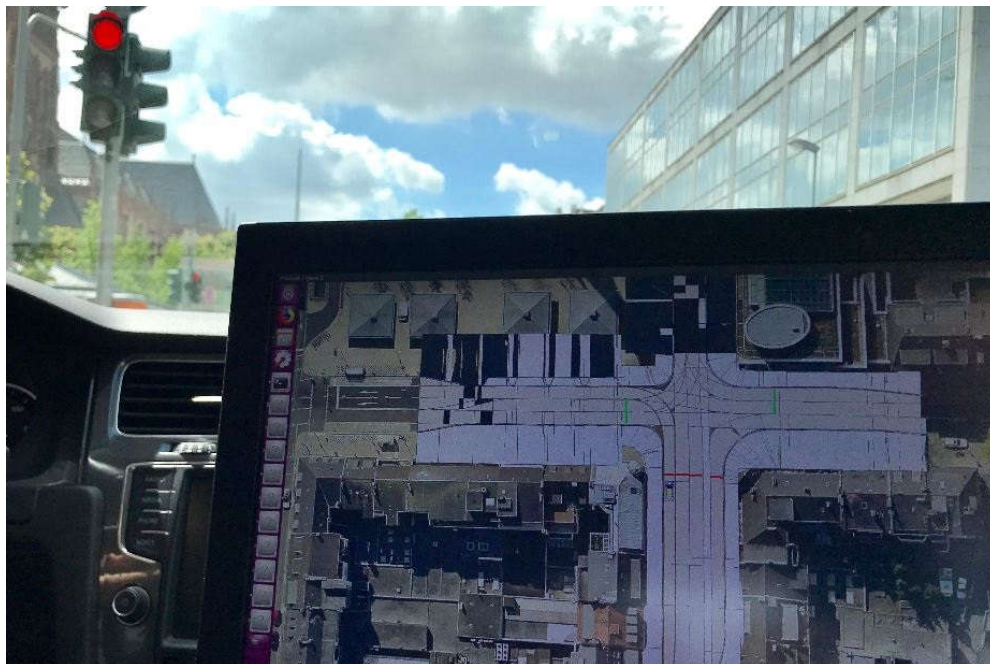


Abbildung 7: Live-Birdeye-Viewer auf dem Testfeld Düsseldorf

Auf dem Abschlussevent am 01.07.2020 in Düsseldorf wurde dann die entwickelte Fahrfunktion erfolgreich einem breiten Publikum und Pressevertretern demonstriert. Dabei wurde das Karree

Friedrichstadt entgegen dem Uhrzeigersinn in automatischer Längsführung befahren, unter Beachtung der Verkehrsteilnehmer (Autos und Personen) der Ampelanlagenstati und -Prognosen sowie unter Nutzung der über Kommunikation in das Fahrzeug eingespielten Daten des Testfeldes.

Siehe zum Beispiel den TV-Beitrag https://www.youtube.com/watch?v=8w_PPXVV8TE

Zusammenfassend demonstrierte das DLR mit dem Forschungsfahrzeug FASCarE das dynamische Einbringen von Zusatzinformationen über multimodale Kommunikation wie zum Beispiel Ampelraten, Daten über Sonderereignisse oder auch strategischen Informationen zum Routing. Unterstützt durch die ebenfalls vom DLR entwickelte digitale Karte des Testfeldes Düsseldorf konnte so abschließend das automatische Fahrmanöver unter Nutzung der Zusatzinformationen demonstriert werden.

1.3 Fahrzeugseitiger Routingclient

Das DLR hat eine Fahrzeug-Client App entwickelt (Tablet oder Smartphone), auf dem der Fahrer unter Anzeige der aktuellen Fahrzeugposition auf einer Routingfähigen Karte durch die Hauptroute des digitalen Testfeldes Düsseldorf (siehe Abbildung 10) geroutet wird. Dabei werden ihm die auf der Strecke befindlichen Schilder mit dWiSta, Infotafeln, VBA entlang Hauptroute, Parkleitinformationen) zur Anzeige gebracht.

Beginn der Hauptroute ist stets die Annäherungsfahrt aus Nord kommend vor dem AK Meerbusch, Endpunkt ist stets eins der drei folgenden Parkhäuser in Friedrichstadt:

- Parkhaus Friedrichstraße, Friedrichstraße 40F, Zufahrt über Herzogstraße
- Parkhaus Luisenstraße (Parkhaus KÖ), Luisenstraße 11, (51.2173° N, 6.7783° W)
- Parkhaus Talstraße, Luisenstraße 33, Breitengrad/Längengrad: (51.2172° N, 6.7797° W)

Der Fahrzeug-Client überprüft kontinuierlich, ob auf der HAV-Schicht eine von der Stadt vorgegebene strategische Ausweichroute vorliegt. Ist dies der Fall, so wird das Fahrzeug auf eine der beiden definierten Ausweichrouten umgeroutet (siehe Abbildung 8). Auf dieser wird dann bis zur zuvor eingegebenen Zieladresse des jeweiligen Parkhauses umgeroutet. Für den Fall, dass die Belegtsituation der Parkhäuser ein anderes Parkhaus vorgibt wird automatisch auf dieses umgeroutet.

Um die fahrzeugseitige App mit Daten zu versorgen publiziert die Stadt Düsseldorf auf dem MDM (Mobilitätsdatenmarktplatz) die folgenden Daten mit dem Ziel bereit, dass der HAV-Server Verkehrsdaten mit die HAV-Schicht per Abonnement die Daten zyklisch (Intervalle oder Push/Pull-Dienst zwischen) abholt und bereithält. Der vom DLR entwickelte fahrzeugseitige Client wird die Daten von der HAV-Schicht abrufen.

- Aktuell gültige Umleitungsrouten (im Datex II Datenmodell für Verkehrs- und Umfelddaten Version 01-00-00 -08 | 2018)
- Statische und dynamische Zustandsinformationen der an der Strecke befindlichen Schilder (dWiSta, Infotafeln, VBA entlang Hauptroute Parkleitinformationen)
- Beleggrade der Parkhäuser und Vorgabe über das anzusteuern Parkhaus im Zielgebiet Friedrichstadt

Die HAV Schicht stellt die gültigen Ausweichrouten und Triggerpunkte (absolute Geokoordinaten) als XML-File zum Abruf durch den fahrzeugseitigen Client des DLR zur Verfügung.

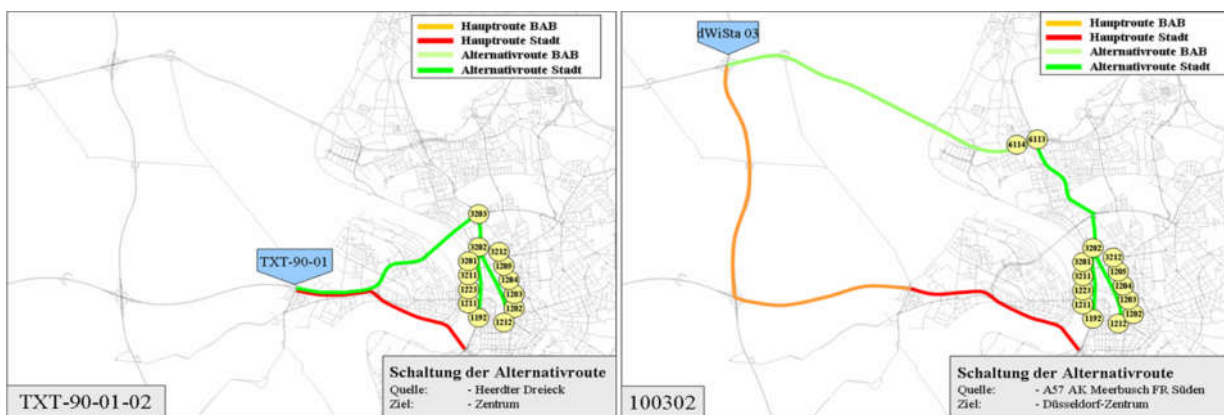


Abbildung 8: Strategische Ausweichrouten der LHD

Die Schnittstellen für diese Funktionalitäten sind in Abbildung 9 dargestellt. Der fahrzeugseitige Routingclient wird über zyklische oder ereignisgesteuerte Abfragen bei der HAV-Schicht über Mobilfunk mit den strategischen Vorgaben der Stadt Düsseldorf versorgt. Die jeweils aktuell vorliegende Datenlage wird dann vom Routingalgorithmus zur Anzeige gebracht.

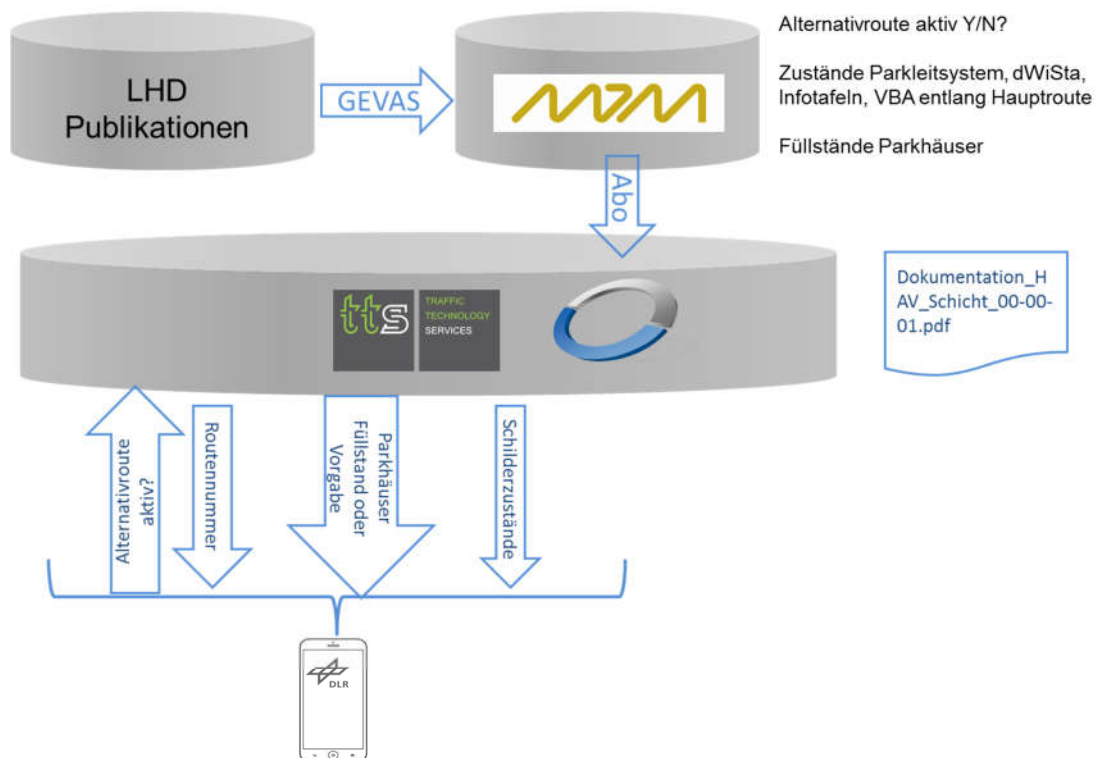


Abbildung 9: Datenfluss Schema für den fahrzeugseitigen Routingclient des strategischen Routings

Das strategische Routing ist durch acht Schnittstellen repräsentiert die im Folgenden kurz genannt werden:

- **DynamicParkingWebservice:** Zur Abfrage des Echtzeitstatus und der Anzahl freier Parkplätze der Parkhäuser.
- **Routing-Webservice:** Zur Abfrage der strategischen Routen, die als Koordinatenliste dargestellt werden. Wenn die strategischen Routen nicht verfügbar sind, ist der gesamte Dienst und damit verbundenen Funktionalitäten der App nicht nutzbar.
- **StaticVarioWebservice:** Anfordern der ID von Überlandverkehrszeichen und deren Positionen.
- **DynamicVarioWebservice:** Zum Abfragen der Echtzeit-Intercity-Verkehrszeichen, die als Bild und Textinformationen dargestellt werden.
- **StaticTunnelWebservice:** Anfordern der ID der Tunnelverkehrszeichen und ihrer Position.
- **DynamicTunnelWebservice:** Zur Abfrage der Bilder der Tunnelverkehrszeichen.
- **StaticOutsideVarioWebservice:** Zur Abfrage der ID der Straßenverkehrsinfos und ihrer Standorte.
- **DynamicOutsideVarioWebservice:** Zur Abfrage der Echtzeitdaten des Straßenverkehrs, die als Bild- und Textinformationen dargestellt werden.

Die zuvor beschriebenen Webservice-Dienste werden im Folgenden in ein Sequenzdiagramm überführt und so chronologisch in ihrer Verarbeitung aufgearbeitet. Abbildung 10 zeigt das Ablaufdiagramm zum besseren Verständnis der App-Arbeitsabläufe.

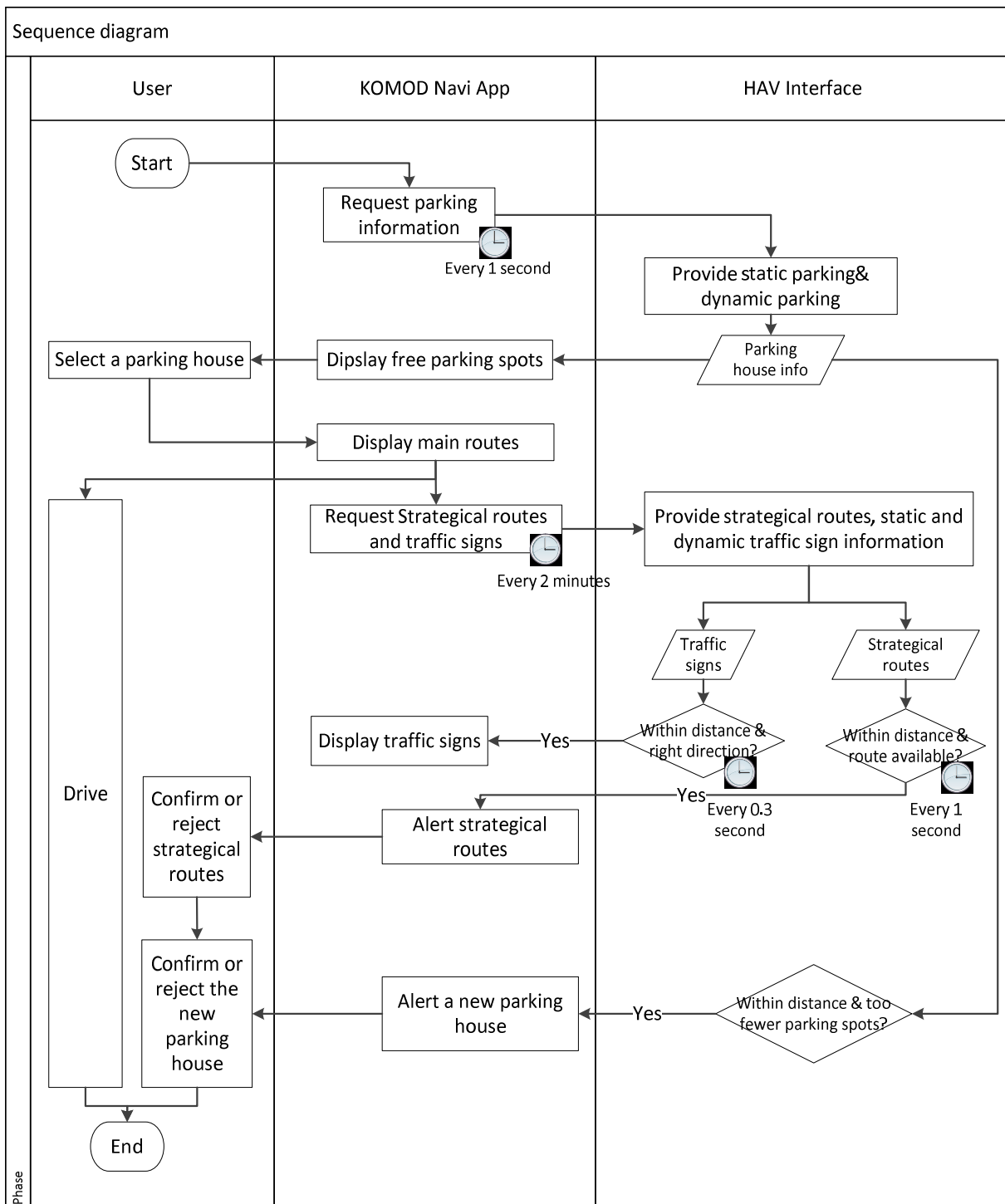


Abbildung 10: Sequenzdiagramm KoMoD Navigation App

Wenn der Benutzer die App startet, fordert die App automatisch die Parkhausinformationen über die HAV-Schnittstelle „DynamicParkingWebservice“ an. Da die freien Parkplätze geprüft werden, wenn sich das Auto dem ausgewählten Parkhaus nähert, wird die Anforderung an die Parkinformationen jede Sekunde ausgeführt. Mit den Parkinformationen kann die App die Parkhäuser und deren kostenlose Parkplätze anzeigen. Der Benutzer kann dann in der Benutzeroberfläche ein Parkhaus auswählen. Nachdem das Parkhaus bestätigt wurde, wird die statische Route vom Startpunkt zum Parkhaus auf der Karte angezeigt und der Benutzer folgt der statischen Route während der Fahrt. In der Zwischenzeit fordert die App automatisch alle 2 Minuten strategische Routen und Verkehrszeichen an, falls sich die Informationen über strategische

Routen und Verkehrszeichen während der Fahrt ändern. Da sich die Informationen jedoch nicht zu häufig ändern, wird die Häufigkeit dieser Anforderungen nur auf 2 Minuten festgelegt. Die relevanten Webservices in der HAV-Schnittstelle sind "DynamicVarioWebservice", "StaticVarioWebservice", "DynamicTunnelWebservice", "StaticTunnelWebservice", "DynamicOutsideVarioWebservice", "StaticOutsideVarioWebservice" und "Routing". Die Details dieser Webdienste werden im letzten Abschnitt erläutert.

Die Verkehrszeichen werden nur angezeigt, wenn sich das Auto innerhalb von 1000 m zu den Verkehrszeichen befindet und die Bewegungsrichtung des Autos mit der Peilung der Straßen übereinstimmt. Daher müssen die Entfernung zu den Verkehrszeichen und die Übereinstimmung zwischen der Bewegungsrichtung des Autos und der Peilung der Straßen überprüft werden. Um ein gutes Timing der Verkehrszeichenanzeige sicherzustellen, wird die Prüfung alle 0,3 Sekunden ausgeführt. Die strategischen Routen müssen alarmiert werden, wenn sich das Auto dem Kontrollpunkt einer strategischen Routen nähert. Daher wird auch die Entfernung zum Kontrollpunkt der strategischen Routen jede Sekunde überprüft. Wenn die Anzahl der freien Parkplätze im ausgewählten Parkhaus weniger als 10 beträgt, wird in der App auch ein Benachrichtigungsfeld angezeigt. Wenn die Warnfelder angezeigt werden, sollte der Benutzer entscheiden, ob er die Vorschläge akzeptiert. Sobald der Nutzer das gewünschte Parkhaus erreicht hat, endet die Abfolge der Sequenzen in der App.

Entsprechend dem Ablaufdiagramm sind die Screenshots der wichtigsten Bedienschritte der App in Abbildung 11 dargestellt.

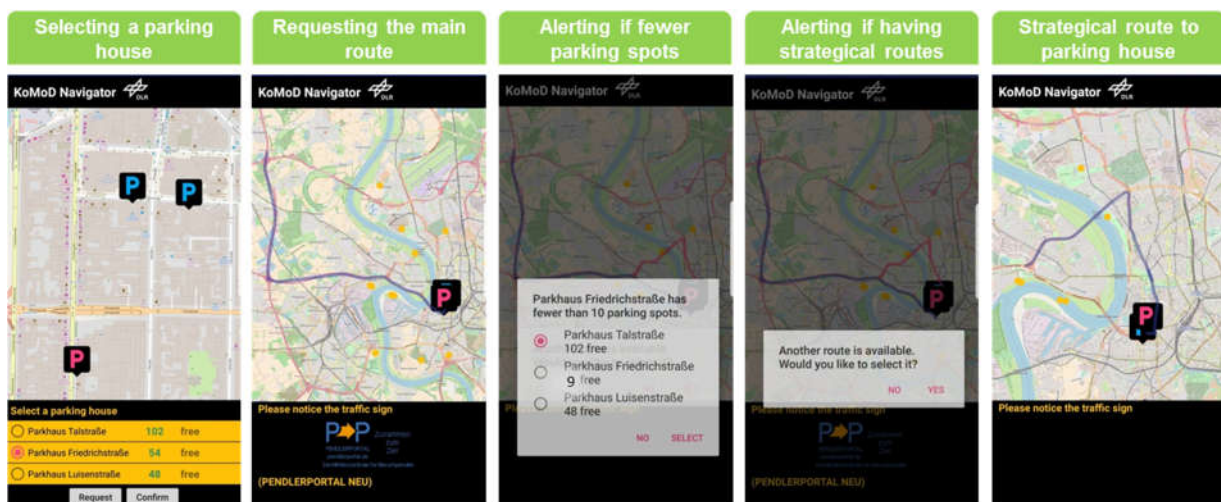


Abbildung 11: Funktionen in der App

Die App beginnt auf der Seite "Auswahl eines Parkhauses". Der Name der Parkhäuser und die Anzahl der Parkplätze werden angezeigt. Wenn dieser Bereich nicht angezeigt wird, dann sind keine freien Parkhäuser verfügbar und der Nutzer drückt nur „Bestätigen“ ohne die Auswahl eines Parkhauses. Wenn die Informationen des Parkhauses nicht korrekt angezeigt werden, kann der Benutzer die freien Parkplätze erneut anfordern, indem er auf die Schaltfläche "Anfordern" klickt.

Der nächste Schritt ist das Anzeigen von Verkehrszeicheninformationen. Die statische Hauptroute ist auf der Karte eingezeichnet und die Orte, an denen die Verkehrszeichen sich befinden, werden ebenfalls auf der Karte angezeigt. Sofern die folgenden Kriterien erfüllt werden:

- Die Entfernung zwischen dem aktuellen Standort und dem Verkehrszeichen ist weniger als 1000 Meter.

- Die Peilung der Straße und die der Fahrtrichtung des Fahrzeugs sind bis zu einem gewissen Grad gleich.

Sequenziell wird das Parkhaus abgefragt. Wenn das ausgewählte Parkhaus weniger als 10 Parkplätze hat, wird ein Benachrichtigungsdialogfeld angezeigt, in dem der Benutzer gefragt wird, ob er ein anderes Parkhaus auswählen möchte (siehe Mitte Abbildung 11). Das Parkhaus mit der maximalen Anzahl von Parkplätzen ist die Standardeinstellung. Die Bedingungen, die den Alarmdialog des Parkhauses auslösen sind:

- Der aktuelle Standort ist nicht mehr als 1000 Meter vom Beginn der letzten Meile entfernt.
- Die freien Plätze des ausgewählten Parkhauses sind weniger als 10.

Ebenfalls sequenziell wird die strategische Route abgefragt. Wenn es eine neue strategische Route gibt, wird ein Benachrichtigungsdialogfeld angezeigt, in dem der Benutzer gefragt wird, ob er seine aktuelle Route ändern möchte. Die strategische Route wird auf der Karte zusammen mit der aktuellen Route angezeigt. Die Bedingungen, die den Alarmdialog der strategischen Route auslösen sind:

- Neue strategische Route vorhanden
- Der aktuelle Standort ist nicht mehr als 1000 Meter von einem strategischen Wechselpunkt der Kreuzung entfernt, wodurch sich die Route ändern kann.

Die Hauptfunktionen der App sind die Auswahl eines Parkhauses und die Anzeige der Verkehrszeichen, wenn die Verkehrszeichen in der Nähe des aktuellen Standorts sind.

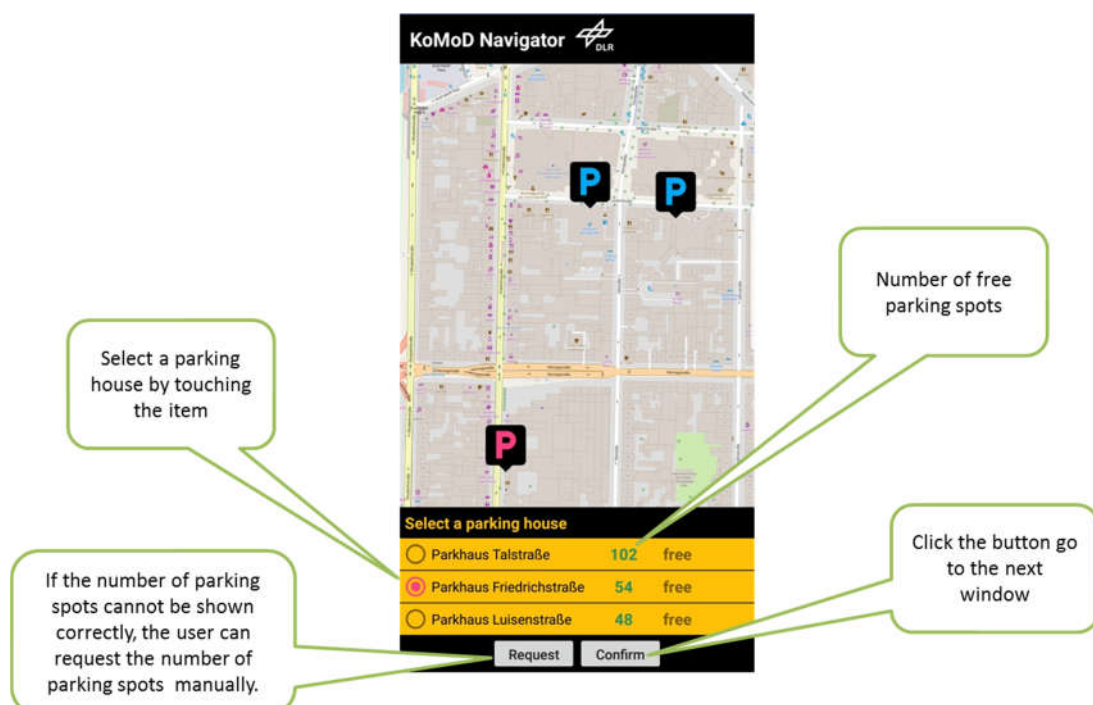


Abbildung 12: Auswahl eines Parkhauses

Die App startet auf der Seite "Auswahl eines Parkhauses" (siehe Abbildung 12). Der Name der Parkhäuser und die Anzahl der Parkplätze werden angezeigt. Wenn die Parkhäuser geschlossen sind, können keine freien Parkplätze ausgewiesen werden. Durch Tippen auf das Parkhaus und Klicken auf "Bestätigen" kann ein Parkhaus für ein Ziel ausgewählt werden. Wenn die In-

formationen des Parkhauses nicht korrekt angezeigt werden bzw. zu aktualisieren sind, kann der Nutzer die Parkplätze erneut anfordern, indem er auf "Anfrage" klickt. Wird ein Parkhaus bestätigt, wird das Parkhausauswahlfragment durch das Fragment der Verkehrszeichenanzeige ersetzt.

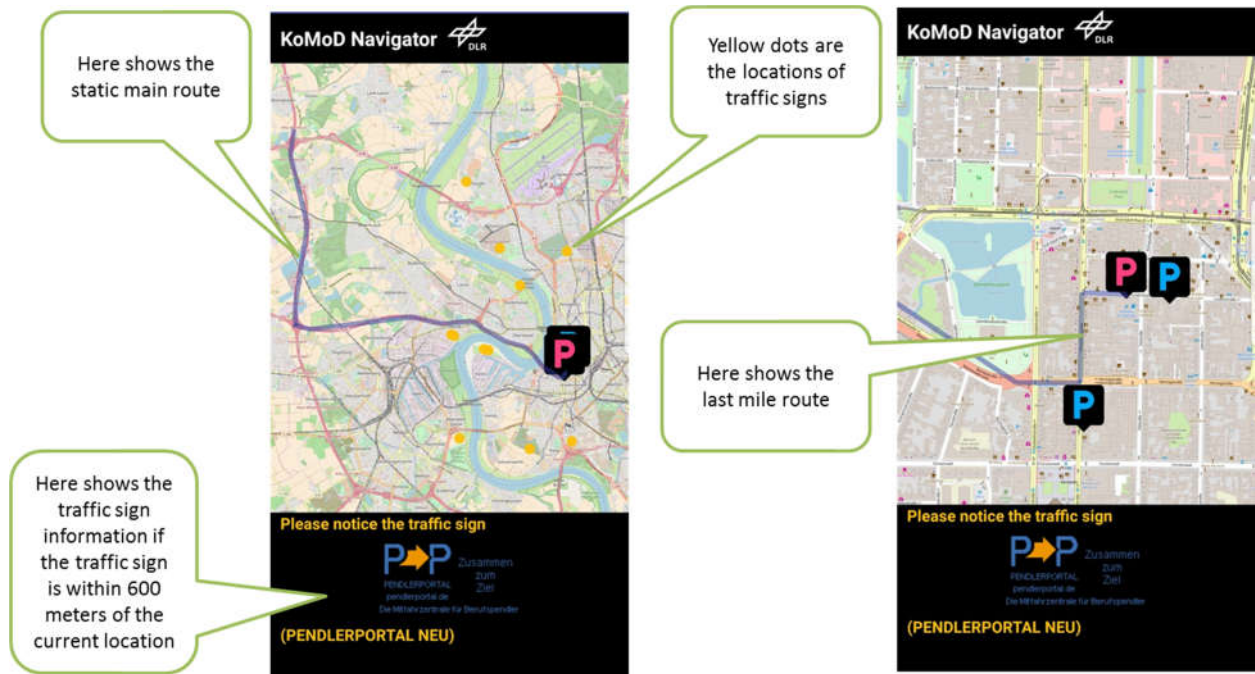


Abbildung 13: Anzeige Verkehrszeicheninformation

Abbildung 13 zeigt das Fragment der Verkehrszeicheninformation. Die statische Hauptroute (blau) sowie die Orte der Verkehrszeichen (gelb) werden auf der Karte gezeichnet. Ein Verkehrszeichen wird automatisch angezeigt, wenn die Entfernung zwischen dem aktuellen Standort und dem Verkehrszeichen weniger als 600 Meter beträgt. Wenn sich mehrere Verkehrszeichen innerhalb der Entfernung von 600 m befinden, wird das nächstgelegene angezeigt. Die Entfernung wird jede Sekunde aktualisiert.

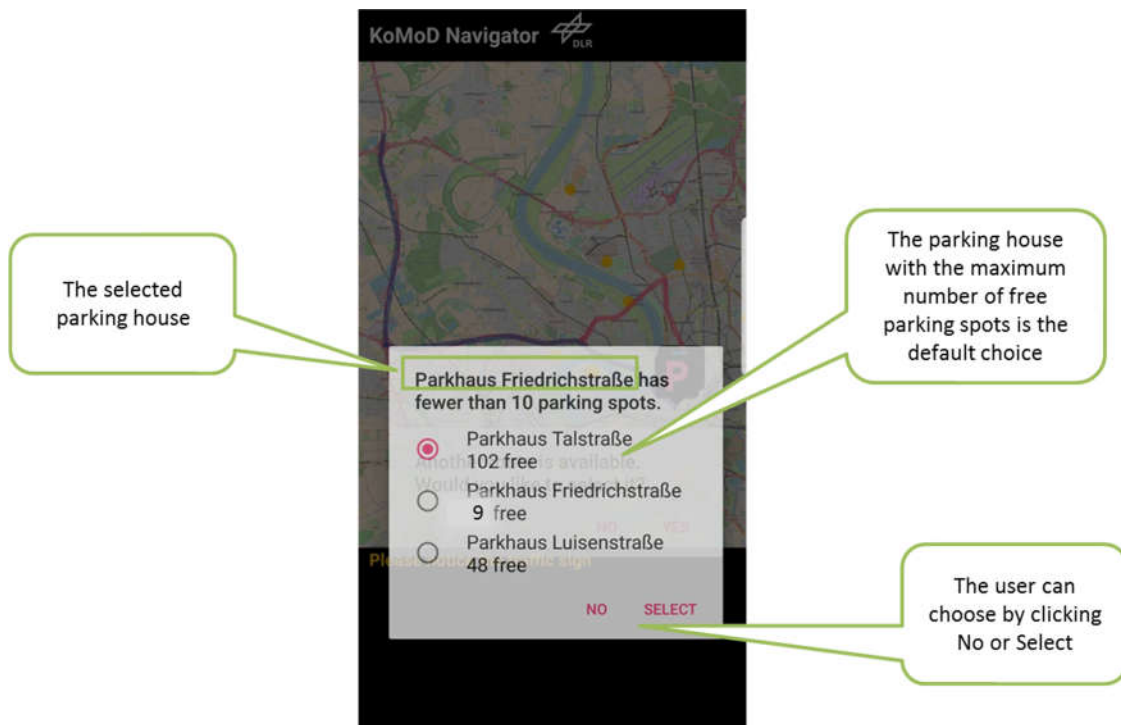


Abbildung 14: Parkhauswarnung

Wenn das ausgewählte Parkhaus weniger als 10 Parkplätze frei hat, wird ein Warndialog angezeigt, der den Benutzer fragt, ob er ein anderes Parkhaus anfahren möchte (siehe Abbildung 14). Das Parkhaus mit der maximalen Anzahl an Parkplätzen ist die Standardwahl.

Die folgenden Bedingungen lösen den Alarmdialog des Parkhauses aus, deren Referenz alle 2 Sekunden geprüft wird:

- Der aktuelle Standort befindet sich innerhalb von 1000 Metern am Ziel.
- Die Anzahl der freien Parkplätze des gewählten Zieles ist kleiner 10.

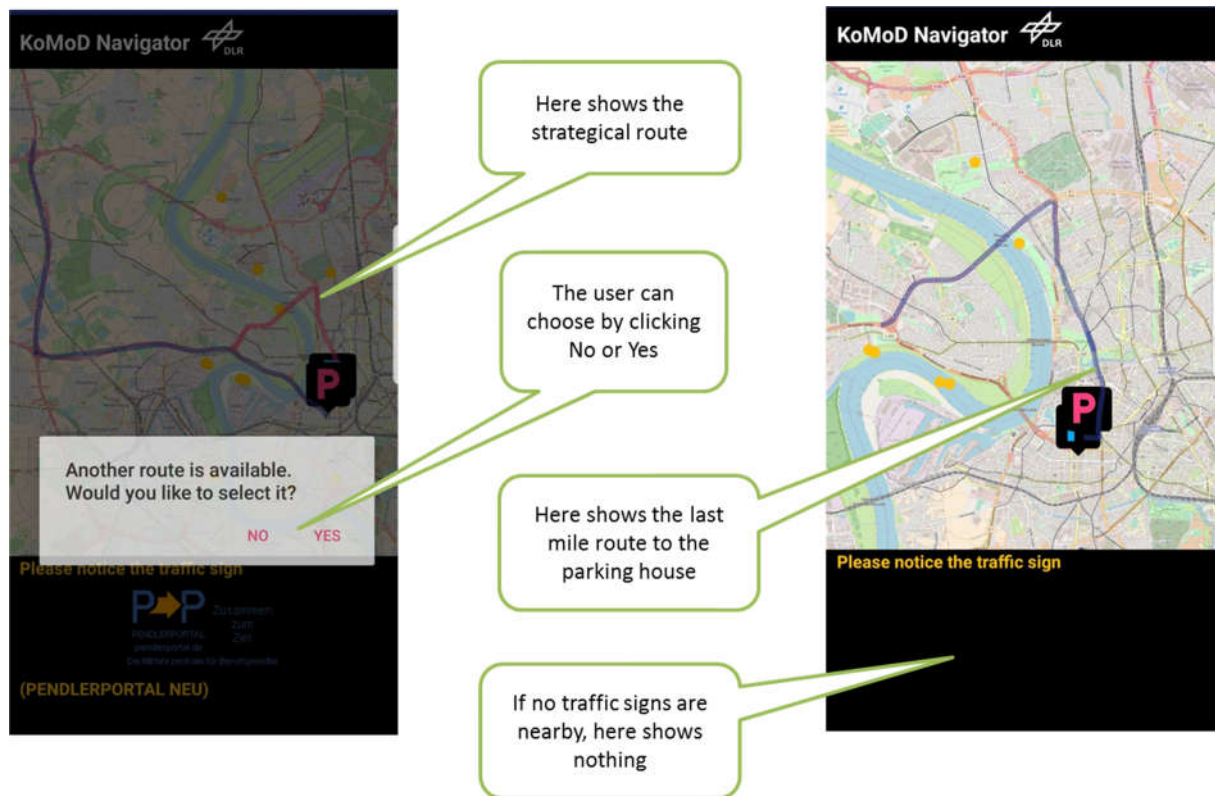


Abbildung 15: Auswahl strategische Route

Des Weiteren wird eine strategische Route angezeigt sofern es eine gibt. Hier wird ebenfalls im 2-Sekunden-Tonus, die entsprechende HAV-Anfrage gestellt. Wenn eine neue Route im System vorhanden ist, wird ein Warndialog angezeigt, der den Benutzer fragt, ob er seine aktuelle Route ändern möchte (siehe Abbildung 15). Die strategische Route wird zusammen mit der statischen Route auf der Karte angezeigt.

Die folgenden Bedingungen lösen den Alarndialog der strategischen Route aus:

- Eine strategische Route ist im System hinterlegt.
- Der aktuelle Standort befindet sich innerhalb von 1000 Metern zum Startpunkt der hinterlegten strategischen Route.

1.4 Digitale Karte (HD-Map)

Eine wichtige Anforderung bei der Automatisierung von Systemen besteht darin, dass die Regelungstechnik der beteiligten Objekte (hier: Fahrzeuge und lokales Verkehrssystem) Kenntnisse über alle relevanten statischen und dynamischen Objekte im aktuellen Manöverraum in Zeit und Ort haben.

Während die dynamischen Objekte (z.B. Autos, Fußgänger) durch Bordsensorik und/oder Echtzeit-Kommunikation in das Fahrzeug eingespielt werden können bietet sich für die Positionen und Art der statischen Objekte im Manöverraum eine hochgenaue digitale Karte an, aus der unter Kenntnis der Egoposition des Fahrzeugs die relevanten manöverleitenden Bebauungen („Features“) ausgelesen und mit der Manöverplanung in Einklang gebracht werden können.

Die HD-Map für automatisiertes und vernetztes Fahren („AVF“) erfordert die folgenden Erweiterungen der bisherigen, vor allem für Navigationsaufgaben benutzten Kartenstandards von Navigationsherstellern:

- Sie muss hochpräzise sein, d.h. die Abweichung der in der Karte beschriebenen Objekte von der globalen Absolutposition darf nicht mehr als 20cm betragen
- Es müssen in der Karte Objekteigenschaften speziell für AVF beschrieben sein, also z.B. Randbebauungen oder Straßenführungselemente die in kompakter und durch Regelungsalgorithmen auswertbarer Form beschrieben sind
- Die Karte muss sich an europa- oder weltweiten Standards zum AVF orientieren, damit sie skalierbare Automationslösungen ermöglicht
- Es müssen Kartenaktualisierungsprozeduren vorgesehen oder zumindest möglich sein, idealerweise durch laufend erhobene Messdaten aus dem Realfeld. Dies erfordert entsprechende Datenverarbeitungsprozesse

Die Anwendungsgebiete der digitalen Karte lassen sich in die folgenden Hauptanwendungsgebiete unterteilen:

- Planung der Fahrtrajektorie durch das AVF-Fahrzeug durch Kenntnis über den möglichen Manöverraum (ohne dynamische Objekte). Dies setzt eine exakte Egoposition voraus, z.B. durch D-GPS und Koppelnavigation
- Rejustage der aktuellen Egoposition durch Vergleich der durch die Bordsensorik erkannten Objekte mit denen der im Manöverraum „vermuteten“ Objekte der digitalen Karte (sogenannte „Optische Relativortung“). Dies setzt eine grobe Kenntnis über die Egoposition des Fahrzeugs voraus (10m Genauigkeit sind hier ausreichend, also genügt normale GPS-Position und Stützung durch Bordodometrie)
- Anwendung der Karte für Road-Maintenance, also automatisches Erkennen von Abweichungen der Karte mit der Realität durch sensorisch passend ausgestattete Fahrzeuge
- Nutzung der Karte für Simulationen von Verkehrsszenarien im mikroskopischen Bereich. Dies können sowohl Verkehrsflusssimulationen (z.B. SUMO, VISSIM) sein, aber auch

Fahrten mit Probanden in Fahrsimulatoren zum Testen z.B. von innovativen Assistenzsystemen oder VR-Komponenten in einem realistischen Umfeld

Um diesen Anforderungen Rechnung zu tragen wurde die HD-Karte des Testfeldes Düsseldorf in die dafür geeignete Formate projiziert. Diese Formate sind für OpenSource Darstellungs- und Konvertierungstools wie zum Beispiel Quantum GIS zugänglich (<https://qgis.org/de/site/>).

Bei den durch das DLR zur Verfügung gestellten Kartenformaten handelt es sich um OpenDRIVE V1.4, übliche Shapefiles sowie DLR-Road2Simulation V1.2. Abbildung 16 zeigt die Prozessschritte die zum Einfahren der HD-Karte angewendet wurden: Das Vermessen der Strecke in Düsseldorf durch hochpräzise Vermessungsfahrzeuge, die Konsolidierung und Konvertierung in übliche Zwischenformate sowie die Nutzungsmöglichkeiten für Verkehrsplanung, Simulation und im Fahrversuch.

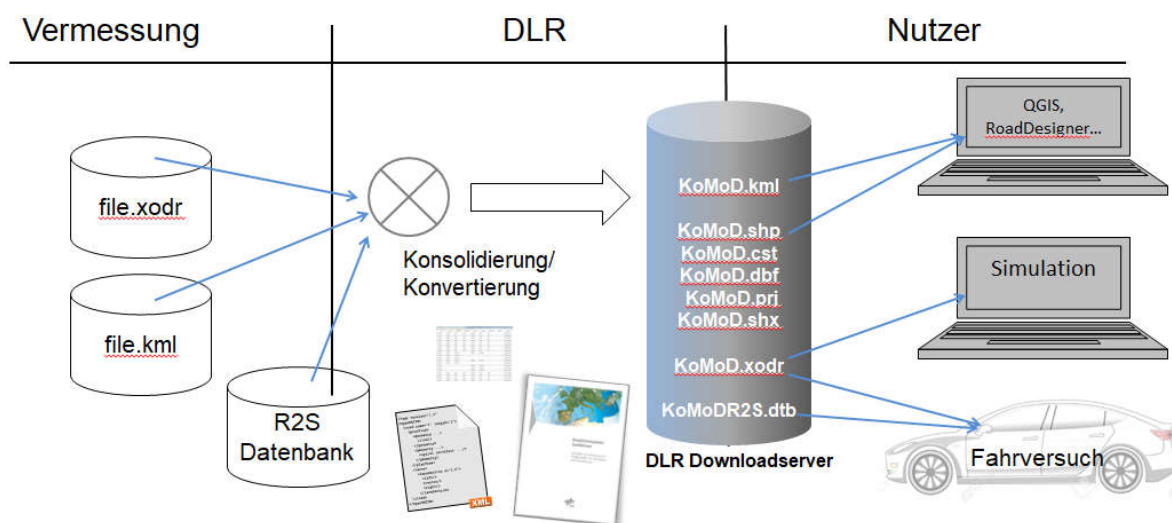


Abbildung 16: Dateiformate der HD-Map: Vermessung und Lieferung, Konvertierung auf dem DLR Geodatenserver, Nutzung in Simulation und Fahrversuch

Allgemeine Erläuterungen zur digitalen Karte des Testfeldes Düsseldorf

Das Hauptformat der HD-Karte Düsseldorf ist OpenDRIVE V1.4, ein OpenSource Format zur logischen Beschreibung von Straßennetzen. Sie werden in Form von XML-Dateien übergeben, die von vielen Konvertierungstools oder Simulationsumgebungen direkt nutzbar sind. Im Falle des Testfeldes Düsseldorf handelt es sich um eine OpenDRIVE Datei, in der das etwa 20 km lange Testfeld vollständig beschrieben ist.

Abbildung 17 zeigt die Werkzeugkette zur automatischen Generierung hochgenauer Karten für die Fahr- und Verkehrssimulation sowie für das autonome Fahren. Die dafür notwendigen Daten werden oft in speziellen Datenformaten benötigt, welche den Straßenverlauf mathematisch abbilden. Diese speziellen Datenformate können OpenDRIVE, RoadXML oder IPGRoad sein. Eine genaue Beschreibung der Werkzeugkette kann aus ([Link](#)) entnommen werden. Die Spezifikation von OpenDRIVE kann aus www.opendrive.com entnommen werden. Die weiteren Spezifikationen und das Hosting des Standards hat die Organisation ASAM (<https://www.asam.net/>) übernommen.

Bei dem Road2Simulation Standard handelt es sich um ein DLR-proprietäres Format, speziell geeignet für Landmarkenrelativierung und Anforderungen des AVF. Da OpenDRIVE sehr auf-

wändig für die Datenerhebung ist, dient Road2Simulation zur vereinfachten Darstellung des Straßennetzes und wurde basierend auf OpenDRIVE entwickelt. Genutzt werden kann es als Zwischenformat und kann mit Hilfe der Werkzeugkette in OpenDRIVE konvertiert werden. Das Hosting des Road2Simulation Standards liegt beim DLR. Die aktuelle Spezifikation kann heruntergeladen werden von:

https://www.dlr.de/ts/Portaldaten/16/Resources/projekte/road2simulation/TD_TS_R2S_Guidelines_v1.2_DE.pdf

Aufbereitung und Bereitstellung der Karte

Verarbeitung der Vermessungsdaten

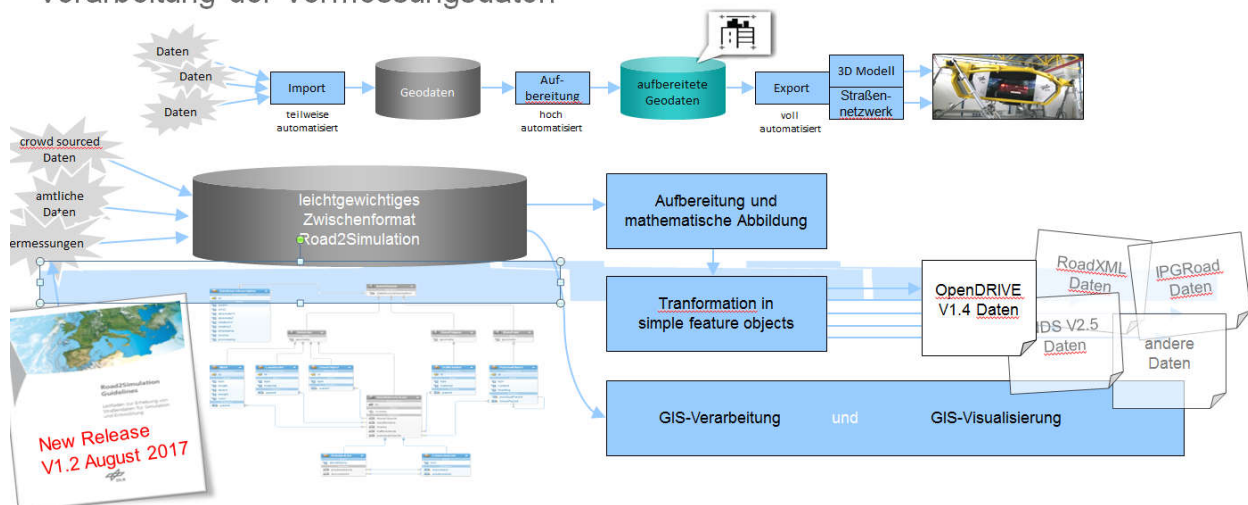


Abbildung 17: Werkzeugkette zur automatischen Generierung hochgenauer Karten

Für die digitale Karte im Testfeld Düsseldorf wurde vom Konsortium die in Abbildung 18 und Abbildung 19 dargestellte Strecke festgelegt. Die Referenzstrecke ist in vier Abschnitte geteilt. A (gelb), A-urban (grün), B (rot) und C (orange):

Abschnitt A: Hauptabschnitt Herzogstraße über Rheinkniebrücke, Rheinallietunnel, Brüsseler Str. bis AS Büberich, beide Richtungen (insg. ca. 13 km).

Abschnitt A-urban: Urbane Kreuzungen und Abfahrten, beide Richtungen (wenn vorhanden):

- Karree Herzogstraße, Elisabethstraße, Fürstenwall, Friedrichstraße,
- Abfahrt Rheinkniebrücke Kaiser-Wilhelm-Ring/Düsseldorfer Str.,
- Abfahrt „Vodafone-Parkhaus“ zur Willstätterstr.

Abschnitt B: A 52 AS Büberich bis AK Karst, beide Richtungen (insg. ca. 12 km).

Abschnitt C: A 57 AK Karst bis AK Meerbusch, beide Richtungen (insg. ca. 15 km).

Die Datenerfassung und -lieferung orientierte sich am Road2Simulation-Leitfaden „zur Erhebung von Straßendaten für Simulation und Entwicklung“¹, um einen gewissen – für die Zielanwendung notwendigen – Mindeststandard zu erfüllen. Das Zielformat der Datenlieferung soll das Straßenbeschreibungsformat OpenDRIVE sein. Für die Vermessung wurden in der Ausschreibung folgende Kriterien (Produktfunktionen) definiert, die sich auf den Road2Simulation-Leitfaden beziehen:

¹ http://www.dlr.de/ts/Portaldaten/16/Resources/projekte/road2simulation/TD_TS_R2S_Guidelines_v1.2_DE.pdf

- Neben dem Straßen- und Spurverlauf mit Markierungen sollen auch relevante Infrastrukturelemente erfasst werden. Zu erfassende Elemente sind:
 - Fahrbahnachsen
 - Fahrstreifenränder
 - Fahrstreifenmarkierungen
 - Fahrstreifenbeschriftungen (bspw. Richtungspfeile)
 - lineare Infrastruktur (bspw. Leitplanken, Lärmschutzwände, Brückengeländer)
 - punktuelle Infrastruktur (bspw. Leitpfosten, Pfosten, Schilder, Schilder- und Mautbrücken)
 - Verkehrsinseln
- Die Autobahnen werden möglichst mit einer `StandardLine` pro Fahrtrichtung modelliert. Auf- und Abfahrten sind mittels `ConnectionLines` zu modellieren.
- Alle Messpunkte der Straßentopographie (im Straßenbereich) dürfen maximal eine relative Abweichung von 50 mm sowie eine absolute Abweichung von 200 mm aufweisen.
- Die Daten müssen in angegeben Datenformaten geliefert werden.
- Die topologische Integrität der Daten muss sichergestellt sein.
- Es müssen keine Oberflächendaten erhoben werden.



Abbildung 18: Übersicht über das digitale Testfeld Düsseldorf. Die markierten Bereiche (gelb, grün, rot, orange) sind die zu konvertierende Referenzstrecke. © 2016 Google, Image Landsat / Copernicus © 2009 GeoBasis-DE/BKG.

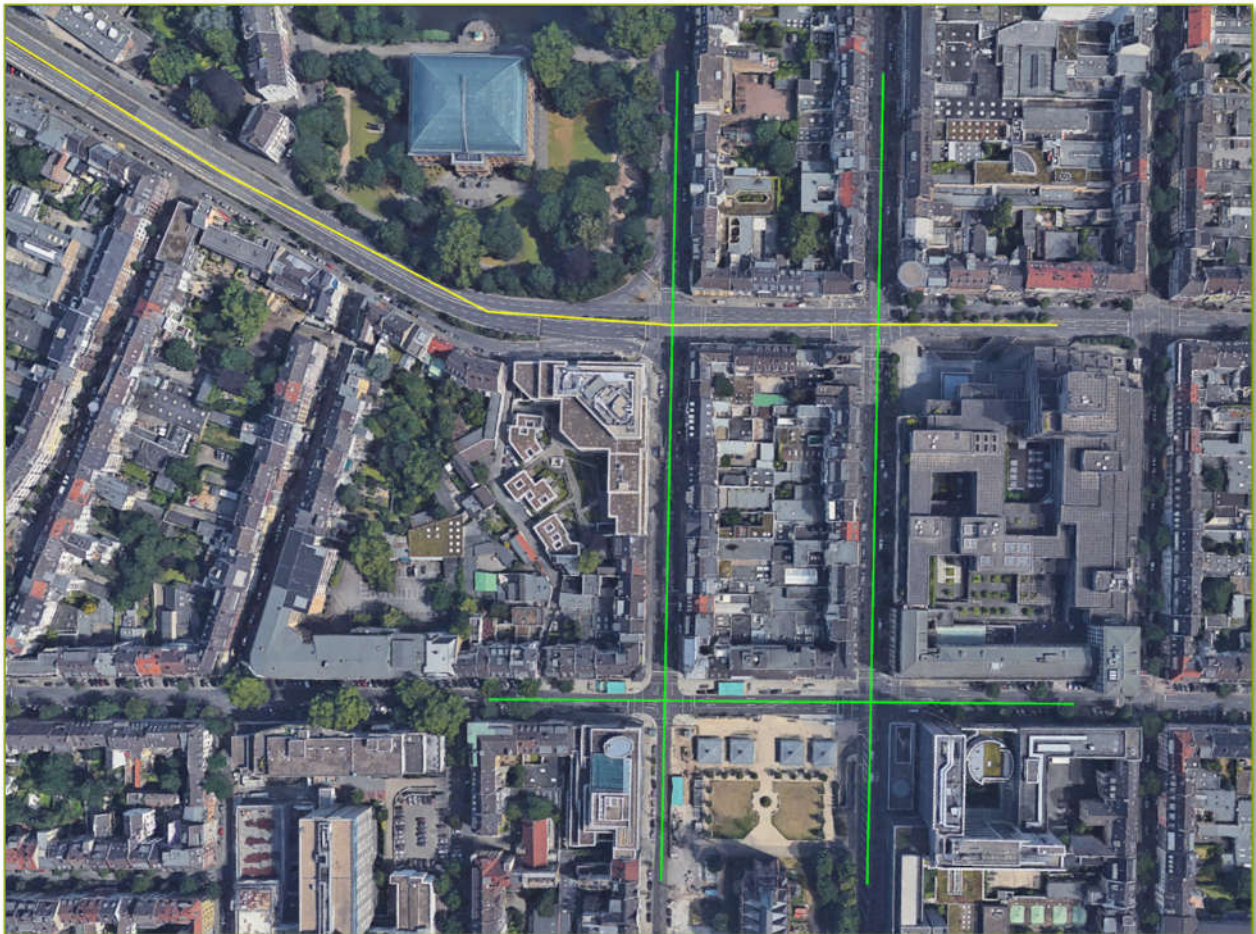


Abbildung 19: Ausschnittsvergrößerung des digitalen Testfeldes Düsseldorf mit dem Teilbereich der urbanen Kreuzungen. © 2016 Google, Image Landsat / Copernicus © 2009 GeoBasis-DE/BKG.

Bereitstellung der Kartendaten über standardisierte OGC-Dienste

Im Zuge der laufenden Vermessung zur hochgenauen, digitalen Karte des Testfelds wird an einer standardisierten Bereitstellungsmöglichkeit der anfallenden OpenDRIVE- und Road2Simulation-Daten gearbeitet. Dies soll einen einheitlichen Zugang zu den Kartendaten und selektive Datenanfragen ermöglichen. Fokus hierbei ist die Wiederverwendung von aus dem GIS-Bereich (Geoinformationssystem) bekannten Werkzeugen, um implizit Zugriff auf die in der GIS-Domäne etablierten Funktionalitäten zu erhalten und „das Rad nicht neu zu erfinden“. Ein weiterer Aspekt ist, dass damit vereinfachter Zugang für Institutionen aus öffentlicher Hand und anderweitige Planungsbetriebe gewährleistet ist, da diese bereits langjährige Erfahrung im Umgang mit standardisierten Geodaten besitzen.

Das vektorbasierte Datenmodell von Road2Simulation abstrahiert die kontinuierlich, mathematisch definierten OpenDRIVE-Elemente bereits zu einer diskreten Darstellung gemäß des OGC-Standards (Open Geospatial Consortium) Simple Feature Access². Alle punktbasierten Geometrielemente werden demnach als `Point`, alle linienhaften Elemente als `LineString` und alle flächigen Elemente als `Polygon` beschrieben. Eine solche Geometriebeschreibung ermöglicht ad hoc das vereinfachte Bereitstellen von Kartenelementen über spezielle Geodaten-Server-Software, wie sie in der GIS-Domäne Verwendung findet. Ein Beispiel dafür ist der etablierte, Java-basierte GeoServer³, der auch im Kontext von KoMoD zum Einsatz kommen soll.

² <http://www.opengeospatial.org/standards/sfa>

³ <http://geoserver.org/>

Der GeoServer wird bei DLR-TS schon zum standardisierten Hosting von Daten anderer Projekte genutzt, ist quelloffen und somit einfach für Zusatzfunktionalität erweiterbar. Der GeoServer basiert auf der umfangreichen Geodaten-Bibliothek GeoTools⁴ und stellt somit automatisch einen breiten Funktionsumfang zum Abfragen, Verarbeiten und Exportieren von Geodaten bereit. Bisher können OpenDRIVE-Daten allerdings noch nicht über den GeoServer bereitgestellt werden. Ziel ist es, durch gezielte Softwareerweiterung standardkonforme OGC-Dienste bereitstellen zu können, die eine Abfrage der Daten über webbasierte Schnittstellen ermöglichen. Ein Beispiel dafür ist die durch den GeoServer zur Verfügung gestellte REST-Schnittstelle.

GeoServer-Erweiterung für OpenDrive-Kartendienste

Im Gegensatz zu diskret definierten Geometrien in Road2Simulation können kontinuierliche OpenDRIVE-Geometrien nicht ad hoc über solch einen Server als Dienst bereitgestellt werden. Eine OpenDRIVE-Datei an sich stellt eine ganze Datenbank mit verschiedenen Geometrietypen (Punkte, Linien, Polygone), geometriellosen Typen sowie deren Beziehungen untereinander dar. Die Bereitstellung eines OpenDRIVE-Datensatzes als Dienst im Sinne des Simple-Feature-Standards erfordert demnach ein Aufbrechen des Datensatzes in seine elementaren Bestandteile, um für jeden einzelnen Elementtyp einen geeignete Karten-Layer erstellen zu können. Ein Layer ist hierbei eine Kartenebene nur eines bestimmten Geometrietyps: Punkt, Linie oder Polygon. Erst dann können die Kartendaten über die beschriebenen Standardwerkzeuge bereitgestellt werden. Aktuell wird an einer GeoServer-Erweiterung gearbeitet, die das Bereitstellen von OpenDRIVE-Daten auf diese Weise als Kartenebenen ermöglicht. Die Erweiterung implementiert einen sogenannten DataStore (Datenspeicher, siehe Abbildung 17) und ermöglicht die Bereitstellung als Kartendienst gemäß des OGC-Standards Web Map Service (WMS)⁵. Durch die Implementierung des Simple-Feature-Standards können ebenfalls direkt Geodaten des OpenDRIVE-Netzes über einen Web Feature Service (WFS)⁶ in weit verbreiteten Geodatenformaten wie Shapefile, GeoJSON oder GML bereitgestellt werden, wenn auch die Beziehung der Elemente untereinander dabei verloren gehen kann, da immer nur Teile einer komplexen OpenDRIVE-Datenbank als eigenständige Layer bereitgestellt werden können.

⁴ <http://geotools.org/>

⁵ <http://www.opengeospatial.org/standards/wms>

⁶ <http://www.opengeospatial.org/standards/wfs>

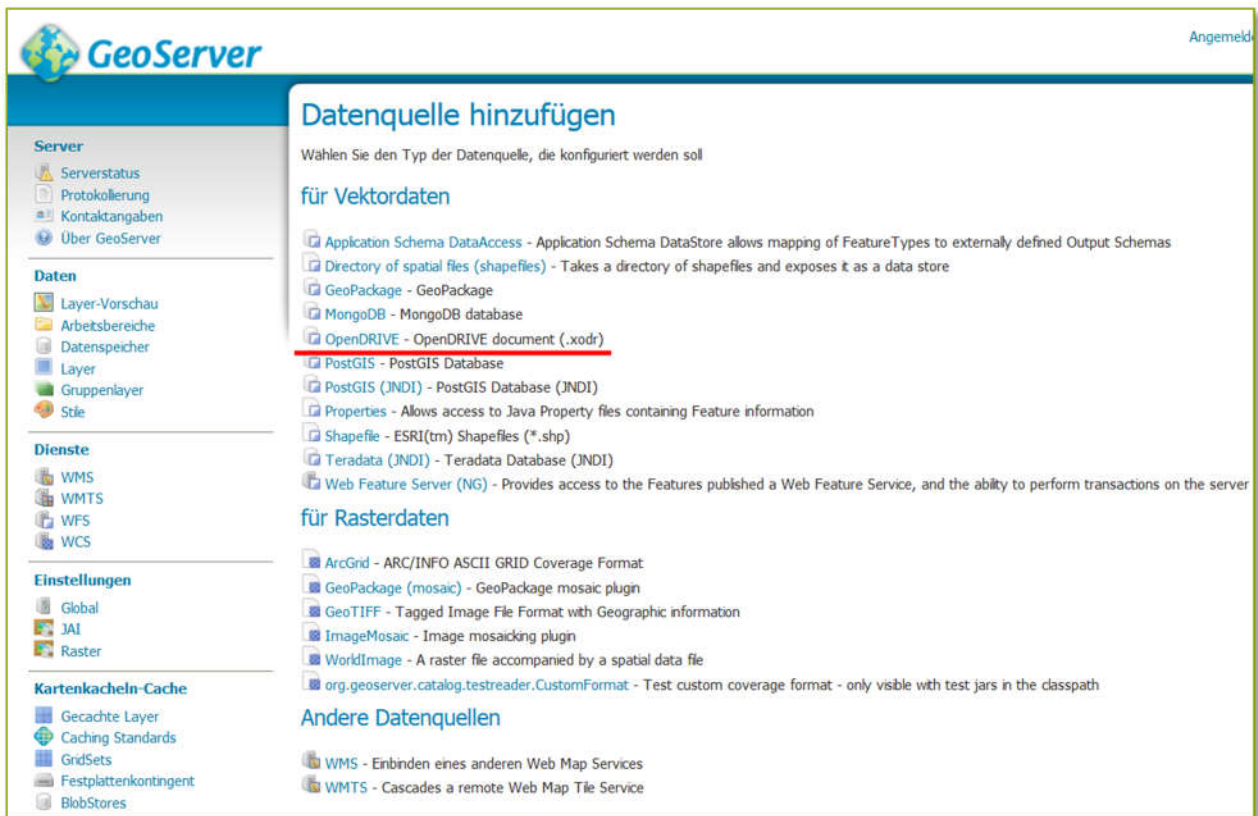


Abbildung 20: GeoServer-Erweiterung zum Bereitstellen von Kartenebenen aus OpenDRIVE-Daten

GeoServer-Erweiterung zur Ausgabe von OpenDRIVE

Neben der zuvor beschriebenen Kartendarstellung von OpenDRIVE-Straßennetzen ist auch die Bereitstellung der eigentlichen OpenDRIVE-Rohdaten gefordert, d.h. die Bereitstellung der gesamten Datenbank mit all ihren Elementen und deren Beziehungen untereinander. Die einfachste Lösung wäre, einen statischen Download-Link zu dem vom Vermessungsdienstleister gelieferten OpenDRIVE-Datensatz zu veröffentlichen. Dies ermöglicht allerdings keinerlei Interaktion mit den Daten in Form von beispielsweise einer Bereichsselektion (Bounding Box) oder Einzelselektion von Straßennetzelementen. Weiterhin soll ggf. eine positionsabhängige Abfrage von OpenDRIVE-Elementen im Umkreis eines Fahrzeugs ermöglicht werden, die nur eine kleine Untermenge des gesamten, mehrere Megabyte großen OpenDRIVE-Datensatzes umfasst. Auch hier kann eine standardisierte Bereitstellung einer Datensatzteilmenge über den GeoServer durch Erweiterung des Web Feature Service⁶ realisiert werden. Da OpenDRIVE nicht zu den durch das OGC standardisierten Ausgabeformaten zählt, ist entsprechend ein neues Ausgabeformat (output format) zu implementieren, siehe Abbildung 20 und Abbildung 21.

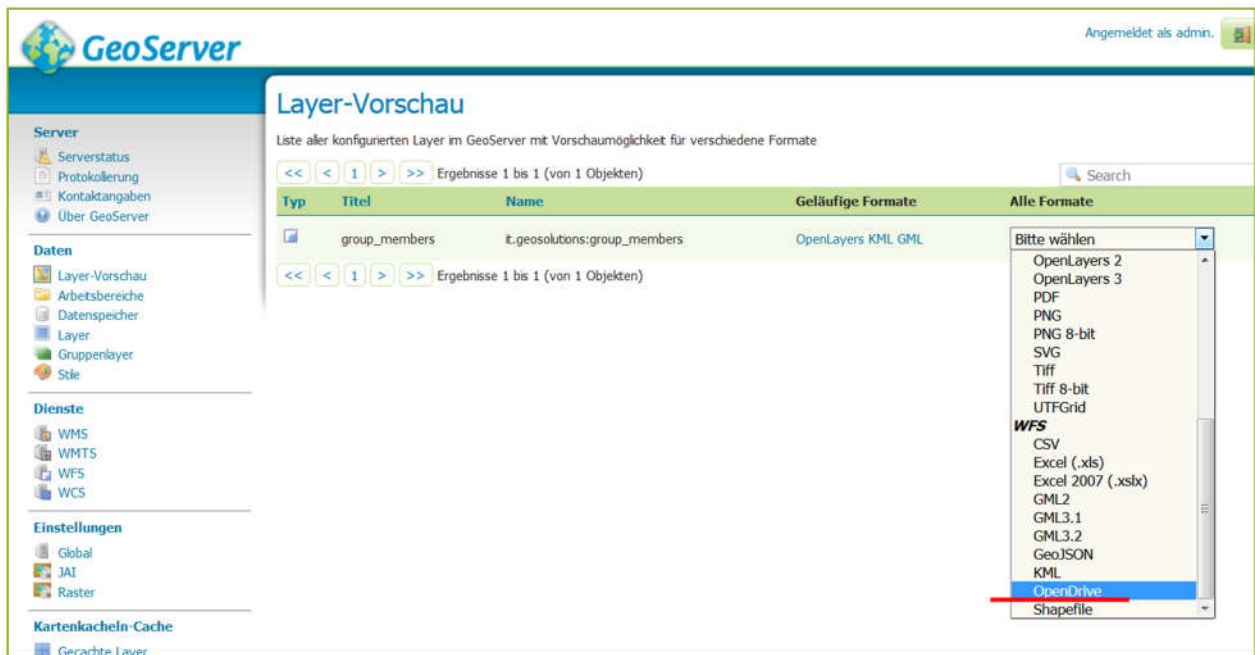


Abbildung 21: GeoServer-Erweiterung um neues Ausgabeformat "OpenDRIVE"

Die Analyse und Nutzbarmachung der HD-Karte ist mit einer Vielzahl von Werkzeugen wie z.B. dem offenem Tool QuantumGIS möglich und erlaubt, je nach Einstellung und Ein/Ausblendung von Layern oder Selektion von Features nahezu unbegrenzte Darstellungsmöglichkeiten. Beispielfhaft sind Abbildung 22 und Abbildung 23 zwei Bereiche im Umfeld des Kreuzungs-Karrées Friedrichstadt dargestellt.

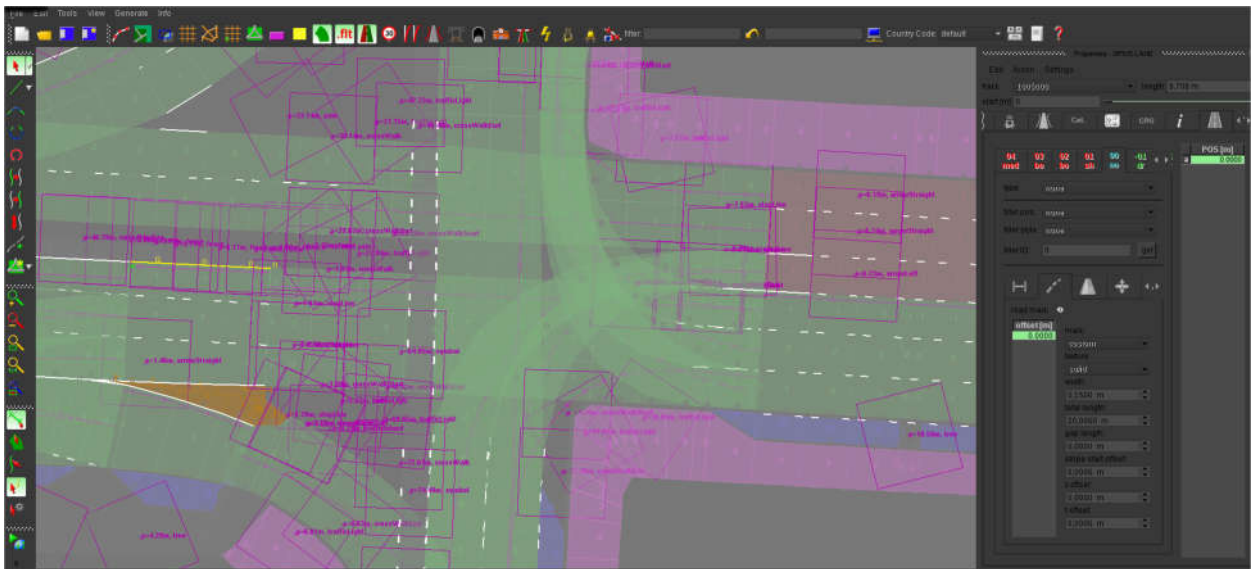


Abbildung 22: OpenDRIVE Karte der Kreuzung Herzogstraße/Elisabethstraße in QuantumGIS

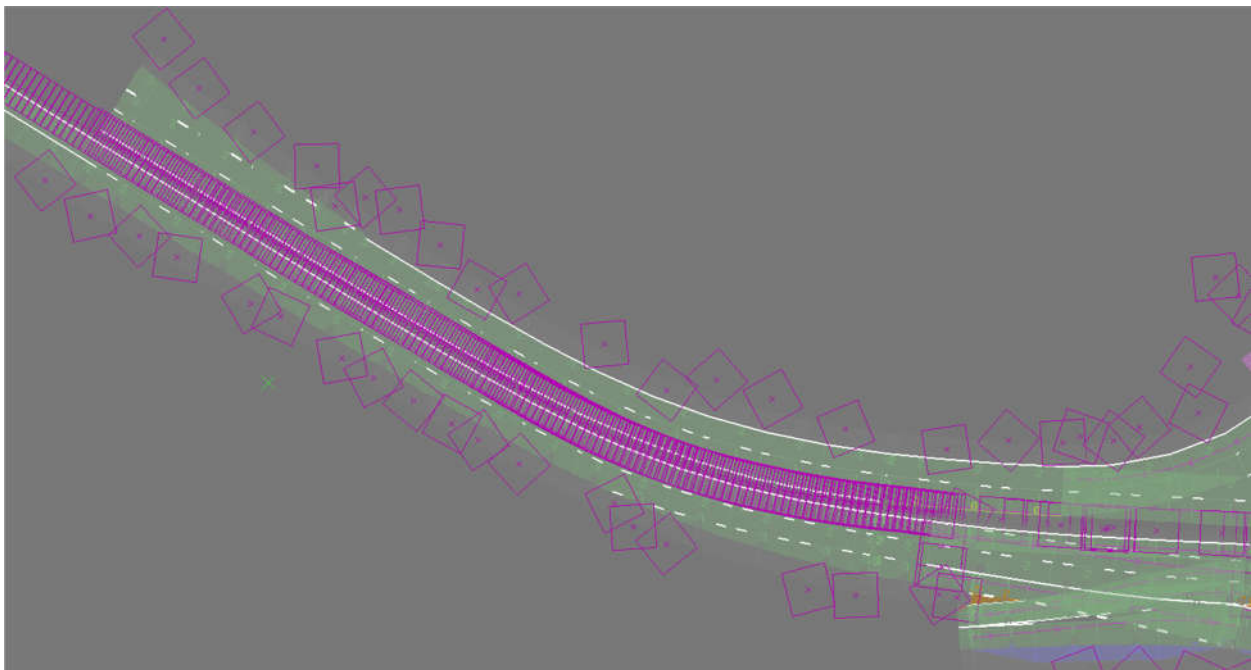


Abbildung 23: OpenDRIVE Karte des zulaufenden Arms auf die Kreuzung Herzogstraße/Elisabethstraße von der Rheinkniebrücke kommend der Kreuzung in QuantumGIS

Berichtsblatt

1. ISBN oder ISSN Nicht zutreffend	2. Berichtsart (Schlussbericht oder Veröffentlichung) Abschlussbericht
3. Titel KoMoD Abschlussbericht KoMoD = Kooperative Mobilität im Digitalen Testfeld Düsseldorf	
4. Autor(en) [Name(n), Vorname(n)] Henning Mosebach, Andreas Richter, Michael Scholz, Jan Lauermann, Qinrui Tang, David Suske, Fabian de Ponte Müller	5. Abschlussdatum des Vorhabens 30.06.2020
	6. Veröffentlichungsdatum 30.01.2020
	7. Form der Publikation Online
8. Durchführende Institution(en) (Name, Adresse) Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR) Institut für Verkehrssystemtechnik Lilienthalplatz 7 38108 Braunschweig	9. Ber. Nr. Durchführende Institution Nicht zutreffend
	10. Förderkennzeichen 16 AVF 1006 H
	11. Seitenzahl 32
12. Fördernde Institution (Name, Adresse) Bundesministerium für Verkehr und digitale Infrastruktur (BMVI) Invalidenstraße 44 10115 Berlin	13. Literaturangaben Keine
	14. Tabellen 0
	15. Abbildungen 21
16. Zusätzliche Angaben Nicht zutreffend	
17. Vorgelegt bei (Titel, Ort, Datum) Nicht zutreffend	
18. Kurzfassung Ziel des Projektes KoMoD bestand in der Nutzbarmachung von infrastruktureseitig vorhandenen und aufbereiteten Informationen des Testfeldes für automatische Fahrmanöver in AVF-Fahrzeugen (AVF=Automatisierte und Vernetzte Fahrzeuge). Das DLR demonstrierte mit dem Forschungsfahrzeug FASCarE das dynamische Einbringen von Zusatzinformationen über multimodale Kommunikation wie zum Beispiel Ampelraten, Daten über Sonderereignisse oder auch strategischen Informationen zum Routing, so dass, unterstützt durch die ebenfalls vom DLR entwickelte digitale Karte des Testfeldes Düsseldorf, automatische Fahrmanöver unter Nutzung der Zusatzinformationen demonstriert wurden. Zu diesem Zwecke wurde die Kommunikationsarchitektur des DLR-Demonstrationsfahrzeugs DLR FASCarE durch Software- und Hardwareanpassungen so an das Datenangebot des Testfeldes Düsseldorf angepasst, dass sowohl über den Kommunikationsstandard ITS G5 als auch über Mobilfunk entsprechend aufbereitete Daten empfangen wurden. Zusätzlich wurden fahrzeugseitige Routing- und Anzeigeapplikationen der Daten des Testfeldes entwickelt und auf Smartphones zur Anzeige gebracht.	
19. Schlagwörter Digitale Verkehrsinfrastruktur, Testfeld, automatisiertes und vernetztes Fahren, Mischverkehr	
20. Verlag Nicht zutreffend	21. Preis Nicht zutreffend

Document Control Sheet

1. ISBN or ISSN n/a	2. type of document (e.g. report, publication) n/a
3. title KoMoD Abschlussbericht KoMoD = Cooperative Mobility on the Digital Testsite Düsseldorf	
4. author(s) (family name, first name(s)) Henning Mosebach, Andreas Richter, Michael Scholz, Jan Lauermann, Qinrui Tang, David Suske, Fabian de Ponte Müller	5. end of project 30th of June 2019
	6. publication date 30.01.2020
	7. form of publication Online
8. performing organization(s) (name, address) German Aerospace Center (DLR) Lilienthalplatz 7 38108 Braunschweig	9. originator's report no. n/a
	10. reference no. 16 AVF 1006 H
	11. no. of pages 32
12. sponsoring agency (name, address) Bundesministerium für Verkehr und digitale Infrastruktur (BMVI) Invalidenstraße 44 10115 Berlin	13. no. of references n/a
	14. no. of tables 0
	15. no. of figures 21
16. supplementary notes n/a	
17. presented at (title, place, date) n/a	
18. abstract The aim of the KoMoD project was to utilize information from the test site Düsseldorf and its infrastructure for automatic driving maneuvers in AVF vehicles (AVF = Automated and Connected Vehicles). With the research vehicle FASCarE, the DLR demonstrated the dynamic introduction of additional information via multimodal communication, such as traffic light data, data about special events or strategic information on routing, so that, using the digital map of the Düsseldorf test field, also developed by DLR, automatic driving maneuvers taking additional information into account has been demonstrated. For this purpose, the communication architecture of the DLR demonstration vehicle DLR FASCarE was adapted by software and hardware adaptations to the data range of the test field Düsseldorf in such a way that correspondingly prepared data was received both via the communication standard ITS G5 and via mobile radio. In addition, vehicle-side routing and display applications for the data from the test field were developed and displayed on smartphones.	
19. keywords Digital Infrastructure, Testsites for automated driving, automated and connected driving, mixed traffic	
20. publisher n/a	21. price n/a