Daniel Burgstahler, Sebastian Zöller, Martin Möbus, Tim Walter and Ralf Steinmetz: Navigate.KOM: Datenbankbasierter Informationsansatz für Fahrassistenzsysteme. In: Proceedings of the 6th GMM-Fachtagung Automotive meets Electronics (AmE 2015) February 2015.

Navigate.KOM: Datenbankbasierter Informationsansatz für Fahrassistenzsysteme

Navigate.KOM: Database-Aided Information Approach for Driver Assistance Systems

Daniel Burgstahler, Sebastian Zöller, Martin Möbus, Tim Walter, Tobias Rückelt und Ralf Steinmetz
Multimedia Communications Lab (KOM), Technische Universität Darmstadt, Germany, Email: Firstname.Lastname@KOM.tu-darmstadt.de

Kurzfassung

Viele auf dem Infotainment System basierende Fahrassistenzsysteme wie beispielsweise das Navigationssystem basieren auf relativ statischen Kartendaten kombiniert mit einer laufenden Datenerfassung verschiedener Fahrzeugsensoren. Zusätzlich wird noch die aktuelle Position, meist per GPS, über eine externe Antenne empfangen. Durch die Kombination mit einer Mobilfunkschnittstelle lassen sich schon heute Zusatzinformationen, wie beispielsweise Staumeldungen, abfragen. Die Funktion selbst, z.B. eine Routenberechnung basiert dann meist auf allen verfügbaren Daten nach einer vom Nutzer eingestellten Priorität, z.B. schnellste Route. Dabei können leider viele weitere Parameter nicht berücksichtigt werden, da bisher keine geeignete Erfassung existiert. So gibt es beispielsweise täglich Veränderungen im Straßennetz in Form von Tagesbaustellen, zusätzlichen teils temporär aufgestellten Verkehrszeichen, Umleitungen etc. Viele neue Fahrzeuge bereits mit Sensorik ausgestattet sind um solche Veränderungen zu erfassen. Unser Ansatz besteht darin diese Daten zu einem Informationsserver zu übertragen. Somit werden die Nutzer dieser erweiterten Informationsbasis nach dem Crouwdsourcing Prinzip auch zu beitragenden Informationsquellen. Auf der Serverseite werden diese Informationen zunächst gefiltert und aufbereitet. Dabei werden Redundanzen genutzt um Zuverlässigkeit und Genauigkeit zu erhöhen. Die so gewonnenen Daten werden in unserem Ansatz über einen Message Broker den Fahrzeugen bereit gestellt. Applikation können über den Message Broker die gewünschten Daten abfragen und als zusätzliche Informationen verwenden. In unserem Aufbau verwenden wir ein Android-Smartphone als mobile Plattform. Das erlaubt eine hohe Flexibilität und macht die Weiterverwendung der Applikationen auch außerhalb des Fahrzeugs möglich. Im aktuellen Aufbau werden Verkehrszeichen zur Datenerfassung von der im Fahrzeug integrierten Kamera erfasst und auf den Fahrzeugbus gesendet. Von dort werden die Daten über einen CAN-Adapter von einem Minicomputer ausgelesen und per Wi-Fi an das Smartphone gesendet. Zur Weiterleitung an den Informationsserver verwenden wir ebenfalls den Message Broker. Die erfassten Daten werden an den Message Broker übertragen und von dort an den Informationssserver weitergeleitet.

Abstract

Most infotainment-based driver assistance systems, e.g., the navigation system, are based on relatively static map data combined with a continuous data acquisition from different vehicle sensors. In addition, the current position is received with an external antenna, commonly by the use of GPS. Additional information, e.g., congestion information, is already available in combination with a cellular communication module. The application feature itself, e.g., a route calculation, is commonly based on all available data according to a user selected priority, e.g., fastest route. However, a lot of further parameters cannot be considered because of so far missing acquisition techniques. As an example, one could mention the daily changes in the road network in terms of roadworks, temporarily erected traffic signs, diversions, and many more. Many modern vehicles are already equipped with sensors that are able to detect such changes. Our approach is to transfer such newly gained knowledge to a central information server. This causes the users of this extended information set also to contribute, similar the crowdsourcing concept. On the server side, the received information is firstly filtered and processed. Redundancy is used to increase reliability and accuracy. For the communication management between the vehicles and the central information server we use a message broker. Vehicles can publish detected information to the broker and request location dependent information as additional information. In our setup we have used an Android smartphone as mobile device platform. This enables high flexibility and also allows to continue the usage of the application outside of the vehicle. In our particular setup, traffic signs are detected by a vehicle built-in camera and the information is transferred to the vehicle data bus. By the use of a CAN adapter, this information is read out by a mini computer and transferred via Wi-Fi to the smartphone. As mentioned before, the data is send to the message broker that relays the data to the information server.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

1 Einleitung

Moderne Fahrzeuge sind bereits heute mit einer Vielzahl an Assistenzfunktionen ausgestattet. Das gilt nicht ausschließlich für Oberklasse-Fahrzeuge, bereits für Kleinwagen sind viele Assistenzsysteme als Option verfügbar. Systeme wir Airbag oder ABS gelten heutzutage als Standard. Das Euro NCAP[1] definiert jedes Jahr eine Liste von Sicherheitsfunktionen welche ein neues Fahrzeug beinhalten muss um eine sehr gute Einstufung zu erhalten. Ein Nebeneffekt ist eine beschleunigte Verbreitung entsprechender Systeme und jedes neue System bringt auch zusätzliche Anforderungen an Hardware, wie leistungsfähigere Steuergeräte und zusätzliche Sensoren, und Software welche im Fahrzeug benötigt wird. In den 90er Jahren hat das CAN-Bus System in die Fahrzeuge Einzug gehalten, welches es Steuergeräten und Assistenzsystemen ermöglicht Informationen und somit Hardware Ressourcen zu teilen. Dies gilt im speziellen für verbaute Sensorik. Ein generelles Ziel ist es verfügbare Systeme und Sensoren möglichst effizient zu verwenden. Die Grenze für solche Systeme liegt jedoch bei der maximalen *Sichtweite* der verwendeten Sensoren. Somit kann immer nur das gegenwärtige Umfeld erfasst und entsprechende Informationen in den Assistenzfunktionen verwendet werden. Für viele Assistenzfunktionen ist es jedoch interessant auch Informationen auf weiter entfernten Wegsegmente zu verwenden. Eine bereits eingesetzte Möglichkeit ist die Erstellung eines so genannten elektronischen Horizonts auf Basis von statischen Kartenmaterial. Allerdings birgt dies auch einige Schwierigkeiten, beispielsweise gibt es im Straßensystem pro Jahr bis zu 15% Veränderungen [1]. Zwar bieten die Kartenhersteller regelmäßige Updates, die Aktualisierung kommt jedoch mit deutlicher Verzögerung. Darüber hinaus ist der Informationsgehalt im vorhandenen Kartenmaterial beschränkt und rein statisch. Insbesondere als Ergänzung zu Navigationssystemen sind auch dynamische Verkehrslagedaten verfügbar. Informationsquellen sind hierfür in der Regel Induktionsschleifen in der Fahrbahn, Sensoren an Autobahnbrücken und vor allem Geräte auf denen diese Daten auch genutzt werden und die über eine Mobilfunkschnittstelle verfügen, wie Smartphones oder Navigationsgeräte. Die Nutzer sind hier also auch die wesentlichen beitragenden Informationsquellen. Dabei gelangen die Informationen über unterschiedliche Technologien zum Fahrzeug. Ältere und einfachere Systeme erhalten die Informationen über den UKW Radio dienst TMC (Traffic Message Channel). Die hierüber verfügbaren Daten sind jedoch stark zeitverzögert und vor allem außerhalb der Ballungszentren relativ ungenau. Modernere Geräte verwenden eine Mobilfunkschnittstelle oder können bereits über DAB+ (Digital Audio Broadcast) entsprechende Informationen empfangen. Die hier zugrunde liegenden Dienste sind meist kostenpflichtig aber auch relativ präzise und aktuell. Durch die inzwischen hohe Verbreitung von Smartphones und vernetzten Navigationssystemen und die Integration verschiedener Datenquellen, wie beispielsweise Sensoren in der Fahrbahn, erreichen die verfügbaren Dienste eine hohe Qualität. Ein starke Einschränkung ist jedoch das diese Dienste nur auf den einen Anwendungsfall hin konzeptioniert sind. Ziel sollte ein System sein das eine Vielzahl verschiedener Anwendungsfälle abdecken kann. Eine andere mögliche Realisierung liegt in der gemeinsamen Verwendung von gesammelten Informationen zwischen verschiedenen Fahrzeugen. Dabei verfolgen wir bei unserem System einen zentralisierten Ansatz mit einem zentralen Informationsserver was mehrere Vereinfachungen ermöglicht. Zum einen ist eine direkte Fahrzeug zu Fahrzeug Kommunikation heutzutage noch nicht verfügbar, wäre aber als Erweiterung in Zukunft bei größerer Verbreitung ebenfalls möglich. Zum Anderen sind nicht nur Ballungszentren in denen von direkt hinter einander herfahrenden Fahrzeugen ausgegangen werden kann von Interesse. In ländlicheren Regionen, außerhalb der Städte und außerhalb stark frequentierter Uhrzeiten ist jedoch damit zu rechnen das Fahrzeuge zwar zeitnah aber nicht unmittelbar hintereinander ein Wegsegment passieren. Somit ist eine Übertragung an ein zentrales System sinnvoll.

Da die zu Verfügung stehende Kommunikationsbandbreite im mobilen Szenario stark variiert und jeder Datentransfer mit Kosten verbunden ist muss die Übertragung von entsprechenden Daten möglichst effizient und zuverlässig realisiert werden. Alle modernen Fahrzeuge sind mit einer großen Anzahl unterschiedlichster Sensoren ausgestattet und können entsprechend ihrer Ausstattung als Informationsquelle beitragen. Es ist also nicht das Ziel das jedes Fahrzeug alle möglichen Informationen sendet sondern entsprechend der Ausstattung einen selektiven Beitrag zum Gesamtsystem leistet. Die von uns vorgeschlagene Kommunikationsarchitektur ist in **Abbildung 1** dargestellt. Für die Kommunikation zwischen den Fahrzeugen und dem zentralen Informationsserver verwenden wir einen s.g. Message Broker. Das ermöglicht eine asynchrone bidirektionale Kommunikation und ermöglicht es auch bei wechselnder Konnektivität Daten vom Informationsserver zu den Fahrzeugen zu senden.

Im folgenden geben wir zunächst eine Übersicht des verwendeten System Setups und der bereits realisierten Funktionalität. Anschließend werden zwei Anwendungsszenarien als Beispiele erläutert. Danach geben wir einen überblick über verwandte Arbeiten. Zum Schluss geben wir ei-

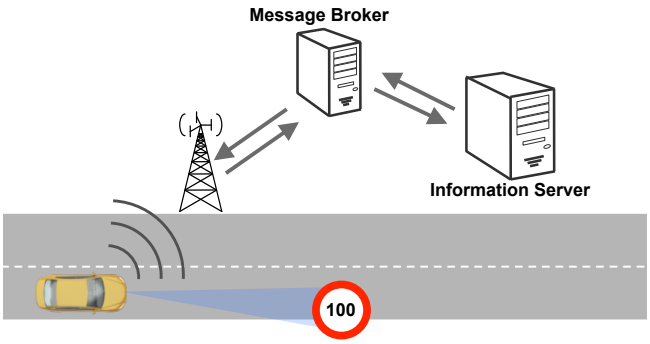

Abbildung 1 Systemübersicht - Beitrag zu zentralem Wissen.

[1] European New Car Assessment Program

ne Zusammenfassung und einen Ausblick auf folgende Arbeiten.

2 System Setup und Funktion

Als Ausführungsplattform für unser System auf der Client-Seite haben wir ein Android Smartphone verwendet. Um einen Zugriff auf den CAN-Bus des Fahrzeugs und somit den Zugriff auf die Fahrzeugsensoren zu ermöglichen haben wir einen Raspberry PI [2] Minicomputer verbunden mit einem PEAK PCAN-USB CAN-Interface [2] verwendet. Das Smartphone und der Raspberry PI wurden per WiFi miteinander verbunden. Alternativ könnte diese Verbindung auch über Bluetooth hergestellt werden. Für die Realisierung der Mobilfunkverbindung haben wir sowohl einen mit LTE ausgestatteten Accesspoint im Fahrzeug verwendet als auch einen UMTS-USB-Stick direkt am Raspberry PI. Die beiden Verbindungsarten wurden nicht gleichzeitig verwendet.

Für die Kommunikation zwischen Fahrzeug und Server-Seite setzten wir einen Message Broker ein. Nachrichten für einen Client, z.B. ein Fahrzeug, werden nicht direkt zu diesem gesendet sondern zum Message Broker. Der Client selbst versucht eine ständige Verbindung zum Message Broker aufrecht zu halten. Wird die Verbindung unterbrochen wird umgehend versucht eine neue Verbindung herzustellen. Trifft eine Nachricht beim Message Broker ein wird diese sofort an den Empfänger weiter geleitet sofern dieser verbunden ist. Alternativ wird die Nachricht unmittelbar nach einem neuen Verbindungsaufbau gesendet. Als Netzwerkverbindung werden TCP Sockets verwendet. Die komplett asynchrone Kommunikation hat den Vorteil im mobilen Szenario sehr robust zu sein und verlagert die Komplexität der Verbindungsverwaltung hin zum Message Broker. Als Implementierung haben wir den *Mosquitto* [4] Message Broker eingesetzt welcher die Open Source Nachfolge des von IBM entwickelten *Really Small Message Brokers (RSMB)* darstellt. Der *Mosquitto* Message Broker ist sehr leichtgewichtig und benötigt nur sehr geringe Hardware Ressourcen. Darüber hinaus wird Authentifizierung und Verbindungsverschlüsselung unterstützt.

Als Nachrichtenprotokoll haben wir *MQ Telemetry Transport (MQTT)* gewählt. Das Protokoll ist sehr leichtgewichtig und verursacht einen sehr geringen Nachrichten Overhead. *MQTT* wurde entwickelt um den Nachrichtenaustausch zwischen kleinen, stark hardware beschränkten Geräten in stark beschränkten Netzwerken zu ermöglichen. Das Protokoll ist optimiert um mit wechselnder und schlechter Verbindungsqualität zurecht zu kommen.

Bei der Wahl des Serialisierungsprotokolls haben wir aus verschiedenen verfügbaren Protokollen wie XML, JSON, Thrift, etc. dann Protobuf [5] gewählt weil es die größte Robustheit bei gleichzeitig sehr kleinen Datenpaketen gewährleistet. Protobuf Datenpakete sindz.B.im Vergleich zu XML bis zu 10fach kleiner und auch die Serialisierung selbst benötigt deutlich weniger Ressourcen, bzw. ist deutlich schneller.

[2] http://www.raspberrypi.org

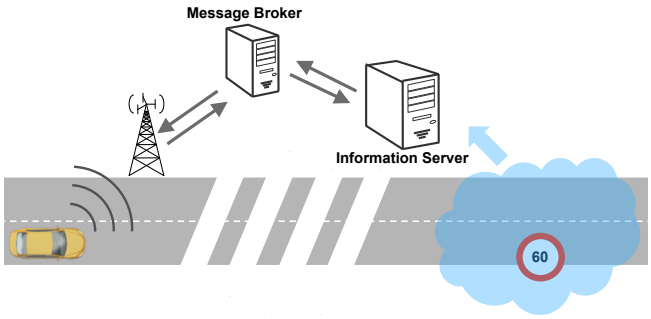

Abbildung 2 Systemübersicht - Abruf von zentralem Wissen.

Auf der Server-Seite verwenden wir eine Java-Applikation auf einem Linux Server welche direkt mit dem Message Broker kommuniziert. Für die Datenverwaltung setzten wir das Objekt-relationale Datenbankmanagementsystem *PostgreSQL*[3] mit der Erweiterung *PostGIS*[4] ein. PostGIS erweitert PostgreSQL um geographische Objekte und ermöglicht lokationsbasierte Abfragen direkt in SQL. Um eine Routenberechung direkt auf der Datenbank per SQL Abfrage zu ermöglichen haben wir die Erweiterung *pgRouting*[5] verwendet. Diese ermöglicht das direkte Ausführen von verschiedenen Suchalgorithmen zur Bestimmung des kürzesten Pfades wie beispielsweise die kürzester Pfad Bestimmung nach dem Dijkstra oder A* Algorithmus.

Als Kartenmaterial haben wir das quelloffene *OpenStreetMap (OSM)* [3] verwendet. Die OSM Karte liegt im menschenlesbaren XML Format vor, was jedoch zu einer sehr großen Datenmenge führt und nicht direkt in PostGIS anwendbar ist. Um die OSM Kartendaten verwenden zu können müssen diese zunächst in die PostgreSQL Datenbank importiert werden. Wir haben hierfür das zu unserem Setup passende Tool *osm2pgrouting*[6] verwendet.

Von den Fahrzeugen empfangene Daten werden zunächst in einer Eingangsdatenbank gespeichert. Aus Platzgründen wird im Folgenden nicht die Datenstruktur der serialisierten Informationen sondern nur der Datenbank des Informationsservers genauer erläutert. Die Struktur der Eingangsdatenbank ist in **Abbildung 3** dargestellt. Jeder Datensatz (*Info*) beinhaltet verschiedene Sensorwerte. Das sind mindestens die Werte für Längen- und Breitengrad für die Position, sowie in der Regel die Fahrtrichtung. Darüber hinaus können weitere Sensorwerte enthalten sein und optional kann die Information als Ereignis (*EventType*) eines bestimmten Typs definiert sein.

In einem Verarbeitungsschritt werden die Daten der Eingangsdatenbank als Informationen in der Datenbank für verarbeitete Informationen abgelegt. In dieser Datenbank werden keine Sensorinformationen verwaltet sondern ausschließlich extrahierte Informationen. eine Solche Information kann beispielsweise ein Verkehrsschild sein. Die Struktur der Datenbank für verarbeitete Informationen ist in **Abbildung 5** dargestellt. In dieser Datenbank existiert

[3] http://www.postgresql.org
[4] http://postgis.net
[5] http://pgrouting.org
[6] http://pgrouting.org/docs/tools/osm2pgrouting.html

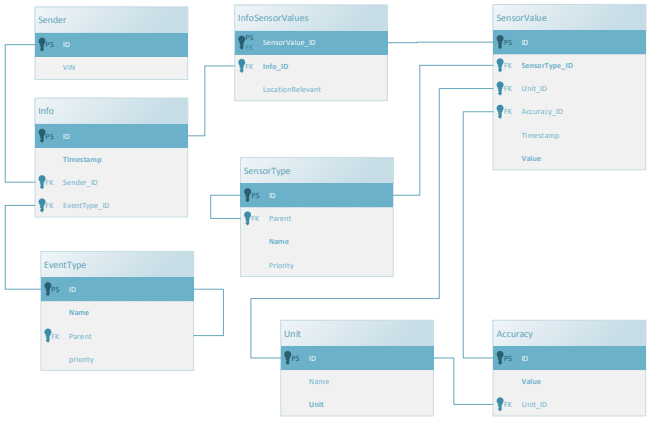

Abbildung 3 Struktur der Eingangsdatenbank.

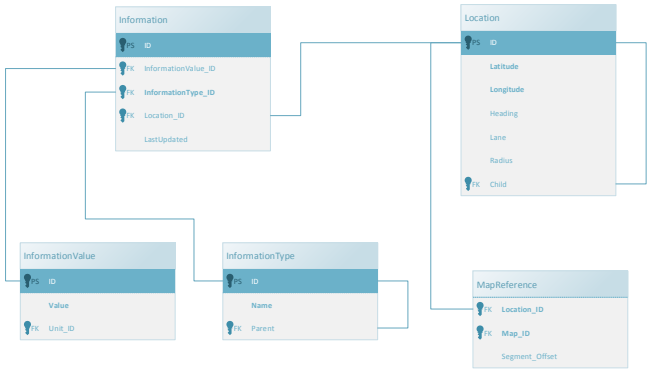

Abbildung 5 Struktur der Datenbank für verarbeitete Informationen.

jede Information nur noch exakt einmal, unabhängig davon wie of dieselbe Information bereits erfasst wurde. Jede Information ist von einem eindeutigen Typ und darf optional einen Wert besitzen. Zwingend ist eine Verknüpfung mit einem Ort (*Location*) der mindestens aus einem Punkt, beschrieben durch Längen- und Breitengrad, besteht. Ein solcher Ortseintrag kann aber auch einen Radius beinhalten um eine Kreisfläche abzubilden. Die einzelnen *Location* Einträge können rekursiv angelegt werden um Wege als Polygonzüge oder freie Flächen als geschlossene Polygonzüge abzubilden. Zusätzlich kann jeder *Location* Eintrag mit einer ID eines Punktes oder Wegsegments aus dem zugrundeliegenden Kartenmaterial, in unserem Fall also der OSM Karte, verknüpft werden.

Der Informationsfluss zur Datenerfassung ist in Abbildung 1 dargestellt. Eine Erfassungsapplikation ließt die Sesordaten vom CAN-Bus des Fahrzeugs und extrahiert daraus die gesuchten Informationen, also z.B. ein erkanntes Verkehrsschild mit dazugehöriger Position und Fahrtrichtung. Diese Information wird dann über den Message Broker an das Serversystem gesendet und dort zunächst in der Eingangsdatenbank gespeichert. Im anschließenden Verarbeitungsschritt wird die neue Information mit dem bereits vorhandenen Datenbestand abgeglichen.

Der umgekehrte Informationsfluss ist in **Abbildung 2** dargestellt. Eine Assistenzapplikation schickt eine Anfrage über eine vorausliegendes Wegsegment an das zentrale Informationssystem. In dem abgebildeten Beispiel hat das System Informationen über eine vorausliegende Geschwindigkeitsbeschränkung welche an das anfragende Fahrzeug zurück gesendet werden.

3 Anwendungsszenarien

Das Spektrum möglicher Anwendungsszenarien ist sehr groß und jede weitere Informationsart, d.h. Sensor oder Ereignisintegration, in einem solchen System bringt wieder neue Anwendungsmöglichkeiten.

Zur Illustration sind im folgenden zwei Beispiele genauer erläutert. Im ersten Beispiel werden Verkehrszeichen erfasst. Mit einer Kamera ausgestattete Fahrzeuge registrieren automatisch Verkehrszeichen und deren dazugehörige Position. Diese Information wird nun an ein zentrales Informationssystem gesendet, wie bereits in Abbildung 1 dargestellt. Da solche Informationen von einer Vielzahl von Fahrzeugen empfangen werden kann in einem Filterungs- und Verarbeitungsschritt sowohl die Genauigkeit der Position verbessert werden als auch eine höhere Zuverlässigkeit sichergestellt werden. Fahrzeuge die später das gleiche Streckenstück fahren können diese Informationen abrufen. Das ermöglicht auch Assistenzfunktionen zum einhalten der aktuell zulässigen maximalen Höchstgeschwindigkeit in Fahrzeugen ohne entsprechende Sensorik. Vor allem aber ermöglicht es auch Informationen bereit zu stellen die außerhalb des aktuellen Sichtbereiches liegen. So kann eine Assistenzfunktion welche die aktuell

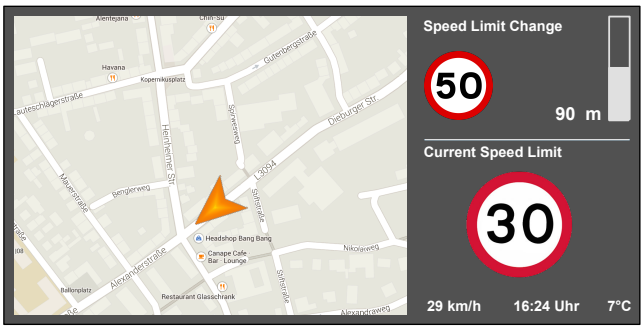

Abbildung 4 Anwendungsbeispiel Verkehrszeichen-Anzeige.

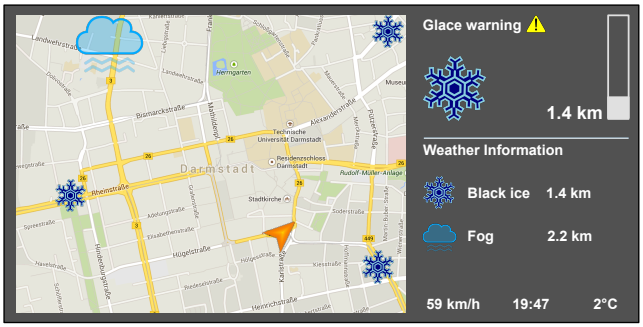

Abbildung 6 Anwendungsbeispiel detaillierte Wetterdarstellung.

zulässige Höchstgeschwindigkeit anzeigt auch bereits die nächste Veränderung der zulässigen Höchstgeschwindigkeit und die dazugehörige Distanz anzeigen. Eine entsprechende Beispielapplikation ist in **Abbildung** 4 dargestellt.
Eine weitere Anwendungsmöglichkeit ist eine feingranulare Wettererfassung. Moderne Fahrzeuge sind mit einer Vielzahl an Sensoren ausgestattet um die Wetterbedingungen zu erfassen. So kann nicht nur die Temperatur oder Luftfeuchte bestimmt werden, sondern beispielsweise auch Glatteis oder die Niederschlagsrate. So kann eine Assistenzfunktion welche die Wettersituation darstellt ein genaues Wetterbild der näheren Umgebung abbilden. Darüber hinaus könnten diese Informationen auch zur optimierten Routenführung und Anpassung der Fahrsituation genutzt werden. Eine entsprechende Beispielapplikation ist in **Abbildung** 6 dargestellt.

4 Verwandte Arbeiten

Das Sammeln und Verarbeiten von Daten unter Einbeziehung einer großen Anzahl an mobilen Geräten wird im Allgemeinen als *Crowd Sensing* bezeichnet. Ganti et al. geben Überblick über mögliche Anwendungsszenarien zur Verarbeitung von Sensordaten mobiler Geräte und definieren *Crowd Sensing* als die Kommunikation einer Vielzahl an Serverapplikationen mit einer Vielzahl an Applikationen auf Mobilgeräten [6]. Dabei wird hier *Crowd Sensing* in drei Kategorien untergliedert: Umgebungs-, Infrastruktur- und Soziale-Anwendungen. die Betrachtung der Verkehrslage und Straßenzuständen und Parkplatzsituationen wird als Infrastrukturapplikationen klassifiziert.
Mit *CarTel* haben Bret Hull et al. ein System entwickelt das Sensordaten von einer Vielzahl von Knoten, z.B. Fahrzeugen, in einem zentralen System, dem s.g. *portal* sammelt [7]. Das System basiert auf einer Vorverarbeitung mit Kompression und Pufferung. Daten können dann per Wi-Fi direkt zum Server oder via Ad-Hoc Kommunikation zu anderen Knoten übertragen werden. Bei CarTel kamen sowohl Smartphone Sensoren als auch Fahrzeugdaten die per OBD2 Diagnose Schnittstelle verfügbar sind zum Einsatz. Der Fokus lag auf der Bereitstellung einer einfachen abfragebasierten API[7] zur Entwicklung von Webbasierten Anwendungen. Die Datenbasis sind in *CarTel* die reinen Sensordaten.
Eine Vielzahl an benötigten Sensoren zur Realisierung von Assistenzfunktionen befinden sich bereits in Smartphones. Hu et al. haben mit SmartRoad ein Smartphone basiertes System entwickelt das Verkehrsregulierende Elemente wie beispielsweise Stoppschilder oder Ampeln erkennen kann[8]. Das System muss nicht mit weiterer Hardware verbunden werden und funktioniert als reine Smartphone Applikation. Erkannte Informationen werden an ein zentrales Serversystem übertragen, verarbeitet und in eine Datenbank gespeichert. Das System unterstützt nur den einen Anwendungsfall, allerdings lag hier der Fokus auf der Erkennung der entsprechenden Informationen.
Ganti et al. haben mit *GreenGPS* ein System entwickelt das es dem Fahrer ermöglicht die Kraftstoff spannendste Route für den verwendeten Fahrzeugtyp zu wählen [9]. Hier wurde Fahrzeugdaten ebenfalls über die OBD2 Diagnoseschnittstelle abgefragt. Aus einem Set von 16 verschiedenen Sensordaten wie beispielsweise Geschwindigkeit, Kraftstoffverbrauch und Position wird die Kraftstoffeffizienz bestimmt. Die gewonnenen Daten werden auf einer SD-Karte gespeichert und müssen nach der Fahrt manuell zum Server übertragen werden. Das System ermöglicht verteilt erfasstes Wissen über Wegsegmente nutzbar zu machen, lässt jedoch die Kommunikation außer acht.
Ein weiteres bedeutendes Anwendungsszenario ist die Parkplatzsuche welche vor allem in großen Städten zu einer deutlichen Erhöhung der Stauhäufigkeit führt [10]. Es wird geschätzt das Staus in den USA allein im Jahr 2007 einen wirtschaftlichen von 78 Milliarden US $ verursacht haben, was die Bedeutsamkeit unterstreicht [11]. Mit *ParkNet* haben Mathur et al. ein System entwickelt das mit Hilfe eines Ultraschallsensors Parkplätze erkennt [10]. Das System bringt zwei wesentliche Beiträge. Zum einen wird eine automatisierte Übersichtskarte von Parkplätzen erstellt, zum anderen werden freie Parkplätze erfasst. Während in *ParkNet* dedizierte Zusatzhardware zum Einsatz kommt verfügen einige moderne Fahrzeuge bereits über eingebaute Parkplatzerkennungssysteme.
Nach unserem Wissen ist unser Ansatz das erste generische System, das völlig Sensor-, Geräte- und Anwendungsunabhängig, Daten in einem modernen effizienten Format überträgt. Für die effiziente Datenübertragung der genannten verwandten Arbeiten kann unser System ebenfalls eingesetzt werden und somit alle Anwendungsfälle abdecken. Einige Erkennungssysteme, wie Beispielsweise die Erkennung freier Parkplätze bei der Vorbeifahrt sind bereits in modernen Fahrzeugen verfügbar und können somit direkt in unser System integriert werden.

5 Schlussfolgerung und Ausblick

Moderne Fahrzeuge sind mit einer Vielzahl an verschiedenen Sensoren ausgestattet welche in unterschiedlichsten Assistenzfunktionen Anwendung finden. Allerdings ist physikalisch bedingt die 'Sichtweite' aller Sensoren begrenzt. Jedoch sind viele weitere Assistenzfunktionen möglich mit Hilfe von Wissen über das Sichtfeld hinaus. Eine mögliche Lösung ist die Bereitstellung eines elektronischen Horizonts als virtuellen Sensor. Ein einfacher Horizont kann bereits aus statischen Kartendaten generiert werden. Durch dynamische Informationen wie beispielsweise Stauinformationen kann dieser erweitert werden. In diesem Beitrag haben wir unseren Ansatz vorgestellt Informationen kollaborativ zu erfassen und zentral zur Verfügung zu stellen.
Das bereits implementierte System ist bisher nur ein einfacher Prototyp unserer Vision eines kollaborativ erfassten elektronischen Horizonts. Als nächste Erweiterung arbeiten wir gerade an einer Server-basierten Horizontgenerierung welche automatisch per Subskription an die Fahrzeugapplikation übertragen und aktualisiert wird. Anschließend ist eine Erweiterung der Vorverarbeitung im Fahrzeug geplant. Eine lokale Teil-Kopie des kollabora-

[7] Programmierschnittstelle

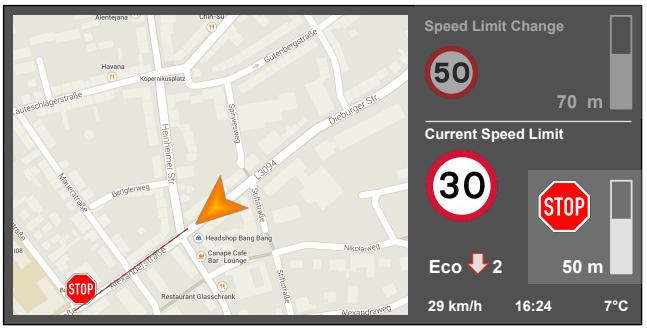

Abbildung 7 Anwendungsbeispiel für einen elektronischen Horizont.

tiv erfassten Modells soll dafür genutzt werden um zu entscheiden ob die erfasste Information einen Mehrwert bietet und übertragen werden soll. Wird so eine bereits bekannte Information wie beispielsweise ein längst bekanntes Verkehrsschild erfasst, so muss diese nicht erneut übertragen werden und die kostenintensive Datenübertragung kann reduziert werden. Außerdem arbeiten wir an einem Konzept um den direkten Informationsaustausch zwischen Fahrzeugen zu ermöglichen. Der zentrale Informationsserver nimmt dann nur noch die Rolle der koordinierenden und überwachenden Instanz ein. Zusätzlich ist eine ganzheitliche automatische Assistenzapplikation geplant wie in **Abbildung 7** beispielhaft dargestellt. Gedacht ist dabei eine Applikation welche kontextabhängig automatisch die aktuell relevanten Informationen anzeigt. Die Anzeige wird dabei je nach Kontextsituation angepasst. So kann zum Beispiel die gegenwärtig zulässige Höchstgeschwindigkeit angezeigt werden und zusätzlich nicht nur die nächste Veränderung sondern auch das nächste relevante Ereignis wie z.B. ein Stoppschild. Auch eine Schaltempfehlung, Abbiegehinweise oder Wetterinformationen können dargestellt werden. Darüber Hinaus soll der kollaborativ erfasste elektronische Horizont als API bereit gestellt werden um eine Basis für zukünftige Assistenzapplikationen zu ermöglichen.

6 Literatur

[1] TomTom, "Kartenaktualisierungsservice." [Online]. Available: http://www.tomtom.com/de_de/maps/map-update-service/

[2] PEAK, "PCAN-USB CAN-Interface für USB." [Online]. Available: http://www.peak-system.com/PCAN-USB.199.0.html

[3] FOSSGIS e.V., "Open Street Map," Online, 2004. [Online]. Available: http://www.openstreetmap.de/

[4] Mosquitto, "An Open Source MQTT v3.1/v3.1.1 Broker," Online. [Online]. Available: http://mosquitto.org/

[5] Google, "Google Protocol Buffers (protobuf)." [Online]. Available: https://developers.google.com/protocol-buffers/docs/overview

[6] R. K. Ganti, F. Ye, and H. Lei, "Mobile Crowdsensing: Current State and Future Challenges," *Communications Magazine, IEEE*, vol. 49, no. 11, pp. 32–39, November 2011.

[7] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "Cartel: a distributed mobile sensor computing system," in *Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, 2006, pp. 125–138.

[8] S. Hu, L. Su, H. Liu, H. Wang, and T. Abdelzaher, "SmartRoad: a crowd-sourced traffic regulator detection and identification system," in *Proceedings of the 12th international conference on Information processing in sensor networks*. ACM, 2013, pp. 331–332.

[9] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "GreenGPS: A Participatory Sensing Fuel-Efficient Maps Application," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 151–164.

[10] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, "Parknet: drive-by sensing of road-side parking statistics," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 123–136.

[11] D. L. Schrank and T. J. Lomax, *The 2007 urban mobility report*. Texas Transportation Institute, Texas A & M University, 2007.

Alle online Referenzen wurden zuletzt im January 2015 überprüft.