

TECHNISCHE UNIVERSITÄT MÜNCHEN
Forschungs- und Lehrereinheit XI
Angewandte Informatik / Kooperative Systeme

Konzept für die Entwicklung eines Echtzeit-Guidings in intermodalen Navigationssystemen (intNAV)

Olga Karsunke

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende: Prof. Gudrun J. Klinker, Ph.D.
Prüfer der Dissertation: 1. Prof. Dr. Johann Schlichter
2. Prof. Dr. Uwe Baumgarten

Die Dissertation wurde am 18.07.2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 31.10.2019 angenommen.

Abstract

This dissertation presents a concept for intermodal navigation systems (intNAV) using smartphone sensors. Although navigation systems are improved over time, there is no seamless navigation for intermodal routes available so far, like it's known in in-car navigation systems. During this dissertation, a solution for an automated identification of a user's transportation mode and correct registration of anomalies in intermodal routes is presented. The results of this thesis are based on five publications, a detailed literature research and two studies. Additionally, the concept for transport mode identification, which was elaborated in this thesis, is patented at BMW in Germany and US. This thesis provides a contribution towards research in the area of transportation mode and anomaly detection in intermodal routes using smartphones.

Kurzfassung

In dieser Dissertation wird ein Konzept für die Erstellung eines intermodalen Navigationssystems (intNAVs) mittels Smartphone-Sensoren vorgestellt. Obwohl die Lösungen für Navigationssysteme immer weiter verbessert werden, ist eine durchgehende Navigation so, wie diese aus einem PKW bekannt ist, noch nicht für intermodale Routen umgesetzt. Dabei gilt es zwei Herausforderungen zu überwinden: Zum einen die automatisierte und korrekte Identifikation des vom Benutzer aktuell verwendeten (öffentlichen) Transportmittels und zum anderen die rechtzeitige Erkennung von Anomalien auf einer intermodalen Route. Die Ergebnisse der vorliegenden Dissertation basieren auf fünf Publikationen, die jeweils Lösungen für die Echtzeit- und Anomalieerkennung auf intermodalen Routen behandeln, auf einer Zusammenfassung der relevanten Literatur zu intNAVs sowie auf zwei empirischen Studien, in welchen Kriterien für intNAVs etabliert und die Funktionalität des entwickelten intNAVs nachgewiesen wurde. Zusätzlich wurde das Konzept für die Transportmittelerkennung, welches im Rahmen der vorliegenden Arbeit entstand, durch BMW in Deutschland und den USA patentiert. Die Ergebnisse der vorliegenden Arbeit liefern einen wesentlichen Beitrag zur Forschung hinsichtlich Transportmittel- sowie der Anomalieerkennung auf intermodalen Routen mittels Smartphone-Sensoren.

Danksagung

Die vorliegende Arbeit wäre ohne die großartige Unterstützung von meinen Lehrstuhl- und BMW-Kollegen sowie von Freunden und meiner Familie nicht möglich gewesen.

Zu allererst möchte ich meinen großen Dank meinem Professor, Prof. Dr. Johann Schlichter, aussprechen, der es mir ermöglicht hat, die vorliegende Forschungsarbeit an seinem Lehrstuhl durchführen zu dürfen, und geduldig meinen Fortschritt kommentierte. Vielen Dank für die Seminare, an denen ich teilnehmen durfte und bei lebhaften Diskussionen meinen Stand der Doktorarbeit bei den Lehrstuhlkollegen vorstellen konnte. Zusätzlich möchte ich Wolfgang und Georg dafür danken, dass ich jederzeit bei euch vorbeikommen konnte, um über den Fortschritt meiner Doktorarbeit zu diskutieren. Das dabei entstandene wertvolle Feedback hat es mir ermöglicht, meiner Doktorarbeit die richtige Richtung zu geben.

Ich hatte das Glück meine Forschungsarbeit bei der *BMW Forschung & Technik GmbH* durchführen zu dürfen. Dies hat es mir ermöglicht dabei von einem großartigen Netzwerk gleichgesinnter Doktoranden und Kollegen zu profitieren. Dazu möchte ich insbesondere meinem *Verkehrsmanagement*-Team Robert, Klaas, Dominik, Christian, Michael, Klaus und Anastasia danken. Es hat mir sehr großen Spaß bereitet, in einem so tollen Team gearbeitet zu haben und von den zahlreichen Diskussionen zu profitieren. Weiterhin geht mein Dank an meine (Ex-)Kollegen Sebastian, Dennis, Benno, Adrian, Hendrik und Maximilian, deren Kaffeeküchendiskussionen eine dankbare Abwechslung zum Dissertationsalltag waren. Ein besonderer Dank geht dabei an Markus, der mit seinen außergewöhnlichen Ideen meine Forschungsarbeit bereichert hatte. Weiterhin möchte ich mich bei meinen Studenten bedanken, deren Ergebnisse ein wichtiger Input für die vorliegende Arbeit sind.

Ich möchte ganz herzlich meinen Freunden und meiner Familie für die großartige Unterstützung während der gesamten Doktorandenzeit danken. Euer Zuspruch und euer Glaube an mich haben manchen Zweifel verfliegen lassen. Ein ganz besonderer Dank gebührt dabei meinem Ehemann Franziskus und meinen Eltern Nadeshda und Peter: Worte können nicht beschreiben, wie dankbar ich euch für eure Liebe und den Glauben an mich bin. Letztendlich danke ich einem kleinen Wesen in meinem Bauch, welches zum aktuellen Zeitpunkt noch gar nicht weiß, wie sehr es mich bei der Vollendung der vorliegenden Doktorarbeit motiviert hat.

VIELEN DANK!

INHALTSVERZEICHNIS

1 Motivation und Aufbau der Thesis	1
Einflussfaktoren auf die urbane Mobilität	1
Informationsdefizite innerhalb einer intermodalen Route	5
1.2.1 Variablen innerhalb (intermodaler) Routen	6
1.2.2 Informationsdefizit - aufgrund fehlender Kommunikation zwischen einzelnen Transportmitteln.....	7
1.2.3 Informationsdefizit - aufgrund fehlender Erkennung des aktuellen Transportmittels	8
1.2.4 Informationsdefizit - aufgrund fehlender Berechnung der Auswirkungen auf die intermodale Route	8
1.2.5 Informationsdefizit - aufgrund fehlender Bewertung der Alternativen	8
Forschungsfragen und wissenschaftlicher Beitrag der vorliegenden Arbeit	9
Aufbau der Thesis	11
2 Grundlagen eines intermodalen Navigationssystems (intNAVs)	13
2.1. Elemente eines intNAVs	13
2.1.1 Begriffsdefinitionen	14
2.1.2 Aufbau eines intNAVs	17
2.2. Transportmittelerkennung mittels Smartphone-Sensoren in Echtzeit.....	21
2.2.1 Transportmittelerkennung mittels einzelner Sensoren	22
2.2.2 Transportmittelerkennung mittels Sensorfusion	30
2.2.3 Transportmittelerkennung mittels zusätzlicher Kontextinformation	32
2.3. Zusammenfassung und weitere Schritte	34
3 Intermodale Transportmittelerkennung (TME) im intNAV	35
3.1. TME mittels Key-points (KP)	36
3.1.1 Konzept von KPs	37
3.1.2 Entscheidungsprozess	40
3.1.3 Weiterführende Konzepte	49
3.2. Datensammlung und –verarbeitung.....	52
3.2.1 Datensammlung	52
3.2.2 Datenmodellierung und digitale Karte	55
3.3 Case Study und Auswertung	59
3.3.1 Prototyp.....	59

3.3.2 Typisches Szenario.....	59
3.4 Zusammenfassung.....	61
4 Prognose von Deviationen im intNAV	63
4.1 Kontext von Deviationen im intNAV	65
4.2 Arten von Deviationen im intNAV	70
4.2.1 Aktive und passive Deviationen	72
4.2.2 Echte und unechte Deviationen	72
4.3 Hypothesen.....	75
4.4 Erkennung des Kontextes von Deviationen	76
4.4.1 Vorhersage der Wahrscheinlichkeit einer Deviation mit Hilfe eines Bayes'schen Netzwerks	77
4.4.2 Deviation in der Raumdimension	77
4.4.3 Deviation in der Zeitdimension	80
4.4.4 Deviationen in der logischen Dimension	82
4.5 Prognose möglicher Deviation.....	89
4.6 Zusammenfassung.....	91
5 Eventgetriebene Architektur für ein intNAV	92
5.1 Theoretischer Hintergrund	93
5.1.1 Eventgetriebene Architektur (EDA)	93
5.1.2 Grundlagen von CEP	95
5.1.3 Event.....	96
5.1.4 Verfügbare Software	97
5.1.5 Esper Details	100
5.1.6 Zusammenfassung.....	106
5.2 Konzept.....	107
5.3 Implementierung	109
5.3.1 Google Directions API.....	110
5.3.2 Esper	111
5.3.3 Prototyp.....	116
5.4 Benchmarktest.....	124
5.5 Zusammenfassung.....	126
6 Umfrage und Nutzerstudie	128
6.1 Umfrage	129
6.1.1 Präferenzen bei der Verkehrsmittelwahl innerorts.....	130
6.1.2 Mediennutzung und Erfahrung mit intNAV.....	131
6.1.3 Bevorzugte Funktionalität eines intNAVs.....	133
6.1.4 Präferenzen bzgl. Inhalt und Stärke der Benachrichtigung bei Anomalien	135
6.1.5 Zusammenfassung.....	140

6.2 Nutzerstudie	141
6.2.1 Ziel der Nutzerstudie	141
6.2.2 Aufbau der Studie.....	142
6.2.3 Auswertung	144
6.3 Interpretation der Ergebnisse.....	153
7 Zusammenfassung	154
7.1 Zusammenfassung der Kapitel.....	154
7.2 Diskussion	155
7.3 Ausblick.....	158
Literaturverzeichnis	160
Abbildungsverzeichnis.....	165
Tabellenverzeichnis	168
Appendix A	169
Appendix B.....	178

1 MOTIVATION UND AUFBAU DER THESIS

Die Mobilität der Zukunft unterliegt einem großen Wandel. Viele Faktoren deuten darauf hin, dass die Art und Weise, wie Menschen zukünftig ans Ziel gelangen, sich ändern wird. Treiber wie der demographische Wandel, innovative Mobilitätskonzepte, neue Technologien und das veränderte Nutzerverhalten führen dazu, dass das Thema **Intermedialität** immer mehr in den Vordergrund gerät. Der Satz „*Nutzen statt Besitzen*“ aber auch „*Mobility on Demand*“ wird immer mehr zum Leitspruch der aktuellen aber vor allem der zukünftigen Generation. Dies hat die meisten klassischen Automobilhersteller dazu bewogen, ihre Mobilitätskonzepte zu überdenken und bereits damit anzufangen, dem Kunden intermodale Lösungen zur Verfügung zu stellen. Zusätzlich sehen sich die Automobilunternehmen durch neue Wettbewerber, welche lange nicht im Fokus standen, bedroht; beispielsweise durch Technologieunternehmen wie *Google*. Technologieunternehmen gestalten mithilfe von neuartigen Mobilitätskonzepten die Trends der Zukunft und zwingen klassische Automobilhersteller zum Umdenken.

EINFLUSSFAKTOREN AUF DIE URBANE MOBILITÄT

Aufgrund der zunehmenden Urbanisierung leben immer mehr Menschen in Städten oder in deren Großräumen. Hochrechnungen der aktuellen World Urbanisation Prospects der UN Bevölkerungsabteilung für das Jahr 2008 ergeben, dass erstmals mehr Menschen in Städten als auf dem Land leben (Swiaczny, 2008). Eine von vielen Folgen dieser Wandlung ist die rasante Zunahme von motorisiertem Individualverkehr in Städten. Deutschland befindet sich weltweit mit einer PKW-Dichte von 532 Autos pro 1000 Einwohner dabei auf Platz zwei, nur noch geschlagen von den USA (statista, 2012).

Um mit dieser steigenden Autodichte umgehen zu können, müssen deutsche Städte ihre Straßeninfrastruktur teilweise komplett neu überdenken. Da die Maßnahmen zur Straßenmodernisierung jedoch viel länger dauern als die Autodichte zunimmt, ist der tägliche Stau auf deutschen Straßen prädestiniert. INRIX ermittelte zum Beispiel die Lerchenauer Straße in München als eine der extremsten Knotenpunkte deutscher Metropolen. Laut dem INRIX Traffic Scorecard Report 2014 steht ein Autofahrer auf dieser Straße durchschnittlich 68 Stunden im Jahr im Stau (Haane, et al., 2015).

Dieser Grund sowie die deutliche Kostenersparnis bewegen immer mehr Menschen in deutschen Ballungsräumen dazu, alternative Verkehrsmittel in Betracht zu ziehen – entweder zusätzlich zum Auto oder als Einzellösung. Hinzu kommt, dass es im Gegensatz zum Kauf und Halten eines eigenen Fahrzeugs deutlich günstiger ist, sich entweder ein Fahrzeug zu teilen oder gleich auf öffentliche Verkehrsmittel umzusteigen. Bei

Letzterem fällt auch die meist langwierige Parkplatzsuche weg, welche in Deutschland im Schnitt zehn Minuten Zeit pro Parkplatzsuche in Anspruch nimmt (Kube, 2013).

Es ist ein Trend zu beobachten, welcher die zunehmende Akzeptanz von alternativen Verkehrsmitteln wie den öffentlichen Personennahverkehr (**ÖPNV**), suggeriert. Im Jahr 2014 sind in Deutschland mehr als elf Milliarden Fahrten mit den öffentlichen Verkehrsmitteln unternommen worden, welches bis dahin einen neuen Rekord darstellte (Reichel, 2015). Zusätzlich zeigt der Bericht des statistischen Bundesamtes auf, dass diese Zahl damit seit zehn Jahren konstant steigt.

Ein weiterer wichtiger Vertreter der Mobilität der Zukunft ist das Car-Sharing. Wo im Jahr 2010 noch knapp 190.000 Nutzer in Deutschland Gebrauch von Car-Sharing Angeboten machten, werden es laut der Trendprognose für "Neue Mobilität" am Fraunhofer Institut schon 1,1 Millionen Nutzer im Jahr 2020 sein. Also ein Anstieg von knapp 480% in 10 Jahren. Interessant ist auch die Quote, mit der sich die Nutzer laut dieser Trendanalyse ein Fahrzeug teilen werden. Im Jahr 2010 kamen auf ein Car-Sharing-Fahrzeug 40 Nutzer. Laut dem vorliegenden Trendreport sollen es im Jahr 2020 schon 60 Personen pro Car-Sharing-Fahrzeug sein (Noeren, et al., 2013). DriveNow¹ oder Car2Go² verzeichnen allesamt kontinuierlichen Zuwachs in der Nutzerzahl. Bayerische Motoren Werke (**BMW**) zum Beispiel, welches mit DriveNow ein Joint Venture zusammen mit Sixt ins Leben gerufen hat, verzeichnete für das Jahr 2016 einen Kundenstamm von 815.000 Menschen – davon allein 607.000 in Deutschland. Dies entspricht einem Kundenzuwachs von 42 Prozent im Vergleich zum Jahreswechsel 2015/2016. Insgesamt sind inzwischen mehr als eine halbe Million Menschen bei DriveNow registriert, welches das Joint Venture dazu bewogen hat, DriveNow in den kommenden Jahren in weiteren europäischen Städten auf den Markt zu bringen (von Nauman, 2016).

Car2Go, welches das Daimler Pendant zu DriveNow von BMW darstellt, fährt einen ähnlichen Erfolgskurs. Die mehr als 1,3 Millionen registrierten Kunden, wovon 793.000 in Europa sind, nahm der Stuttgarter Automobilkonzern zum Anlass neben dem Smart auch künftig ihre B-Klasse für das Car-Sharing anzubieten (car2go, 2016). Neben Car2Go oder DriveNow gibt es noch diverse weitere Car-Sharing-Anbieter - sowohl von weiteren Automobilhersteller als auch von unabhängig geführten Unternehmen - welche ein eigenes Portfolio aufzuweisen haben. Im Rahmen der vorliegenden Thesis ist zum Thema Car-Sharing noch wichtig zu wissen, dass die Nutzerzahlen weltweit seit 2006 stetig zugenommen haben und im Jahr 2014 bei knapp 5 Millionen Nutzern lagen (2014). Die Prognose von den Beratern Frost & Sullivan für die Car-Sharing-Nutzerzahl lag für das Jahr 2013 bei 3 Millionen, womit sie sogar noch pessimistisch kalkulierten, wenn man die Zahlen mit denen des Statistikportals vergleicht. Bis 2020 ist laut der Prognose von Frost & Sullivan vom Jahr 2013 mit nahezu 26 Millionen Car-Sharing-Nutzern zu rechnen (Feick, 2013).

Einer der Gründe, weshalb viele Automobilhersteller wie BMW, Daimler oder Volkswagen eigene Car-Sharing-Lösungen auf den Markt bringen, ist die zukünftige Antriebsform, nämlich die Elektromobilität (Schalk, 2014). Aufgrund der meist kurzen Strecken ist Car-Sharing geradezu prädestiniert für Elektrofahrzeuge. Im Durchschnitt

¹<https://de.drive-now.com/>, aufgerufen am 05. Januar 2016

²<https://www.car2go.com>, aufgerufen am 05. Januar 2016

fährt zum Beispiel ein Car2Go-Kunde mit einem Fahrzeug 20 bis 40 Minuten pro Fahrt (car2go, 2016). Die Reichweite von Elektrofahrzeugen ist im Moment deutlich begrenzt, weshalb sich Elektrofahrzeug in naher Zukunft als Car-Sharing-Fahrzeuge anbieten werden.

Eines der wohl größten Neuerungen der nächsten Jahre im Mobilitätsumfeld ist das hoch- und in naher Zukunft das vollautomatisierte Fahren, welches auch als autonomes Fahren bekannt ist. Beim hochautomatisierten Fahren (**HAF**) kann das Fahrzeug basierend auf der Information, welches es über verschiedene Sensordaten bekommt, Teile einer Strecke selbständig fahren. Jedoch ist der Faktor Mensch beim HAF ständig vorhanden und kann (und soll) im Notfall eingreifen können. Beim autonomen Fahren jedoch ist der menschliche Eingriff nicht erforderlich und so kann der Fahrer des Fahrzeugs sich - ähnlich wie in einem Taxi - vom Auto fahren lassen mit dem einzigen Unterschied zum Taxi, dass es keinen expliziten Taxifahrer gibt, sondern das Fahrzeug sich selbst fährt. Nicht nur traditionelle Fahrzeugbauer wie BMW, Daimler oder auch Volvo arbeiten an Lösungen, welche das Fahrzeug dazu befähigen können, führerlos zu fahren. Automobilferne Firmen wie Google oder Apple arbeiten ebenfalls an hochautomatisierten und autonomen Fahrzeugen. Neben der weiteren Ausstattung solcher hochautomatisierten Fahrzeuge (HA-Fahrzeuge) mit Sensorik und Anpassung der Algorithmen ist das weitestgehend *Vernetzen* solcher Fahrzeuge eines der wichtigsten Zukunftsthemen. Nur wenn herstellerübergreifende Vernetzung und Informationsaustausch der HA-Fahrzeuge stattfinden, können neben dem autonomen Fahren auch Gefahren- oder Staumeldungen aber auch Baustellen frühzeitig kommuniziert werden, damit andere HA-Fahrzeuge, welche diese Information in *Echtzeit* bekommen, diese Stellen „intelligent“ umfahren können. Sobald diese Fahrzeuge serienmäßig fahren und die verkehrsbedingten Rahmenbedingungen geklärt sind, wird das sich überall zu findende Mantra „*Nutzen statt Besitzen*“ immer mehr äußern. So wird in Zukunft das selbstfahrende Auto gerufen, um zu einem bestimmten Ziel zu fahren. Sobald dieses Fahrzeug feststellt, dass es erheblich mehr Zeit kostet so weiterzufahren, wird es die Möglichkeit aufzeigen, mit einem alternativen Verkehrsmittel - zum Beispiel der U-Bahn - weiterzufahren.

Als wichtigster Zukunftstrend gilt jedoch die Tatsache, dass die zukünftige Form der Fortbewegung die Intermodalität sein wird, also das Verwenden von *mehreren* und *unterschiedlichen* Verkehrsmitteln innerhalb einer Route (Hunsicker, et al., 2012). So ist es nicht abwegig, dass eine Person zum Beispiel morgens von zu Hause nicht ausschließlich nur mit ÖPNV oder Auto oder Fahrrad losfährt, sondern diese Verkehrsmittel miteinander - abhängig vom Verkehrsaufkommen - kombiniert. Viele Menschen, insbesondere die, die weit pendeln müssen, nutzen diese Art der Mobilität bereits. Jedoch unterliegt die Intermodalität einigen Informationslücken, wodurch diese Form der Mobilität für viele Benutzer noch vielen Unsicherheiten unterliegt, sodass sie noch nicht selbstverständlich für beliebige Strecken genutzt werden kann. Die angesprochenen Informationslücken werden im weiteren Verlauf dieses Kapitels behandelt. Dabei ist die wichtigste Erkenntnis des Reports von Hunsicker et al. und weiterer Reports folgende:

"Das ideale intermodale Konzept der Zukunft bedeutet für den Nutzer, jederzeit und möglichst spontan die Wahl zwischen öffentlichem Fern-, Regional- und Stadtverkehr, Pkws oder Fahrrädern auf Leihbasis (ggf. mit Elektroantrieb) sowie weiteren Angeboten wie Taxis, Mitfahrzentralen oder herkömmlichen Mietwagen zu haben."
(Hunsicker, et al., 2012)

Martin zu Nedden, Leiter des Deutschen Instituts für Urbanistik, erklärt, dass der mehrfache Wechsel zwischen verschiedenen Transportmitteln in Zukunft Normalität sein werde (Loose, 2013).

Dass Intermodalität eine immer wichtigere Rolle einnehmen wird, sieht man vor allem dort, wo man es am wenigsten erwarten würde. Immer mehr Automobilhersteller beziehen plötzlich für ihre Routenplanung andere, firmenfremde, Transportmittel mit ein. BMW beispielsweise liefert mit der Einführung des i3 eine reichweitenbasierte Karte, welche den Radius des i3's anzeigt. Wenn der Benutzer nun eine intermodale Route im i3 auswählt, dann bekommt dieser eine Routenoption angeboten, welche anzeigt, bis zur welcher Ladestation dieser mit dem i3 fahren kann, um von dort dann zum Beispiel mit DriveNow oder den ÖPNV weiterzufahren. Als Benutzer kann man so flexibler auf kürzere Reichweiten von Elektrofahrzeugen reagieren. Mit *moove*³ bringt Daimler eine Lösung auf den Markt, welche eine intermodale Route ausgibt, nachdem der Start und das Ziel in einer Applikation eingegeben worden sind. Dabei wird der ÖPNV genauso berücksichtigt wie das eigene Fahrzeug, Car2Go oder das Taxi.

In **Abbildung 1** sind die gerade beschriebenen Faktoren für die Mobilitätstrends nochmal dargestellt. Diese Übersicht ist einem Bericht des ADAC's zum Thema „Die neue Form multimodaler Vernetzung“ entnommen worden (Meyer, 2013). In diesem Bericht findet man noch weitere Einflussfaktoren auf die Mobilität der Zukunft wie zum Beispiel die Alterung der Gesellschaft oder das Umweltbewusstsein der Menschen. Diese und die bereits erwähnten Faktoren begünstigen alle das Thema Intermodalität, welche die beste Möglichkeit darstellt, all die erwähnten Bedürfnisse der Menschen und die Trends der Mobilität zu vereinbaren. All diese Neuerungen, welche die aktuell genutzte Mobilität beeinflussen, führen dazu, dass Intermodalität in Zukunft die Lösung für viele Benutzer darstellen wird.



Abbildung 1: Einflussfaktoren auf die Mobilität der Zukunft (Meyer, 2013)

Obwohl sämtliche Trends und Einflussfaktoren in der Mobilität darauf hindeuten, dass mithilfe von Intermodalität sehr viele Vorteile der modernen Mobilität ausgenutzt werden können und sich auch die

³<https://www.moovel.com>, aufgerufen am 31.Januar 2016

Einstellung der Personen positiv zum flexiblen Benutzen von Verkehrsmitteln hin wandelt (vgl. (Loose, 2013)), verlässt sich die Mehrzahl der Personen nach wie vor auf nur ein Transportmittel für ihre Route. Die Gründe hierfür sind vielfältig.

Zunächst besteht das Hauptproblem darin, dass aktuell für das Planen einer intermodalen Route verschiedene Planungssysteme verwendet werden. Dadurch verliert der Benutzer den *Überblick* über die verschiedenen Optionen und kann *nicht flexibel* auf Änderungen reagieren. Auf der einen Seite existieren die traditionellen Auto-Navigationssysteme (**AutoNAV**), auf der anderen Seite gibt es verschiedene Anwendungen, welche zum Beispiel den Fahrplan des ÖPNVs anzeigen.

Man betrachte folgendes Szenario:

Eine Person fährt jeden Tag mit dem Auto ungefähr zur gleichen Zeit in die Arbeit. Auf der Strecke kommt es zum Stau. Nun stellt die Person fest, dass es auf dem Weg in die Arbeit einen Park and Ride (P+R) gibt, von dem aus S-Bahnen zum Ziel fahren.

*Diese Information hilft der Person **aber nur bedingt** weiter, denn es fehlt noch der zweite Teil einer erfolgreichen intermodalen Routenplanung, nämlich die Echtzeitinformation über den **Status der S-Bahn** auf der gewählten Route, inklusive dem **eigenen Status** auf der intermodalen Route.*

*Daher wäre es gut zu wissen, ob die S-Bahn pünktlich ist **und** ob es auf dem weiteren Verlauf der intermodalen Strecke zu Problemen kommt **oder** ob es besser wäre, weiter mit dem Auto durch den Stau zur Arbeit zu fahren.*

Um diesbezüglich nun eine qualifizierte Aussage tätigen zu können, müsste der Status sämtlicher Verkehrsmittel auf der aktuellen intermodalen Route einer Person in Echtzeit bekannt sein. Viel wichtiger ist jedoch die Information darüber, wo sich aktuell der Benutzer befindet, insbesondere dann, wenn dieser mit dem ÖPNV unterwegs ist. Genau in diesem Punkt steckt nun das Hauptproblem von intermodalen Navigationssystemen und liegt der Forschungsschwerpunkt der vorliegenden Thesis. Innerhalb der intermodalen Route herrschen nämlich *Informationsdefizite*, welche die vorliegende Thesis beheben möchte. Dabei geht es nicht nur darum mithilfe von GPS die genauen geographischen Koordinaten festzustellen, sondern vielmehr um die semantische Bedeutung dieser rohen GPS-Daten.

INFORMATIONSDERIZITE INNERHALB EINER INTERMODALEN ROUTE

Ein traditionelles AutoNAV liefert stets *Transparenz* bezüglich der aktuellen Lage auf der Straße. Falls Störungen auftreten, wie zum Beispiel die Bildung eines Staus oder dass der Benutzer sich verfährt, zeigt das System diese Störungen - meistens zuverlässig - an. Zusätzlich bekommt der Benutzer die Information bezüglich der Auswirkung auf seine Route zusammen mit den Alternativen angezeigt. Der Benutzer kennt somit bei dem traditionellen AutoNAV stets den **Soll-** und den **Ist-Zustand**, die **Auswirkungen** und die **Alternativen**. Genau dieses Verhalten fehlt nun innerhalb einer intermodalen Route. Es fehlt also ein Pendant zum AutoNAV, nämlich das *intermodale Navigationssystem (intNAV)*.

Die Erwartungshaltung gegenüber einem intNAV ist dabei die gleiche wie gegenüber einem PKW-Navigationssystem. Dies bedeutet, dass der Benutzer bei der Verwendung des intNAVs ebenfalls stets über den

Soll- und den Ist-Zustand, als auch über die Auswirkungen und die Alternativen auf der intermodalen Route in Echtzeit informiert werden möchte. Intermodale Routen besitzen jedoch viel mehr Angriffspunkte (Informationsdefizite) als eine Route, welche nur mit einem Transportmittel, zum Beispiel dem Auto, befahren wird. **Abbildung 2** stellt dabei die Hauptquellen für das Informationsdefizit innerhalb intermodaler Routen dar. Diese werden nachfolgend (Kapitel 1.2.2 – 1.2.5) detailliert besprochen.

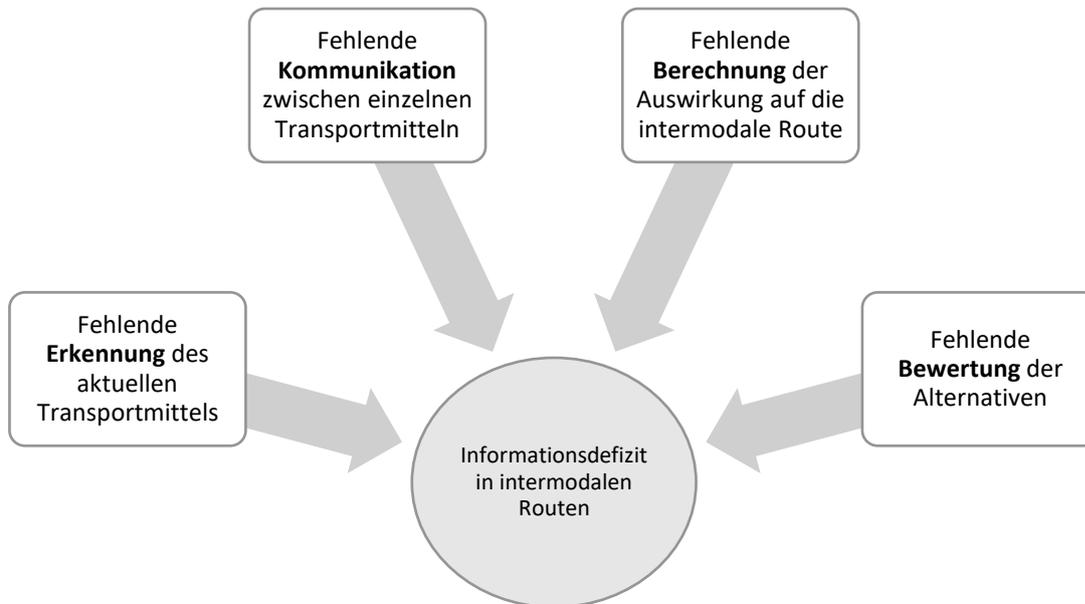


Abbildung 2: Informationsdefizite innerhalb intermodaler Routen

1.2.1 Variablen innerhalb (intermodaler) Routen

Bei einer Route, egal ob intermodal oder nicht, gibt es stets die Variablen „*Transportmittel*“ und „*Benutzer*“. Bei Routen, welche ausschließlich mit dem Auto befahren werden können (also nicht intermodal), kommen die Variablen *Transportmittel* und *Benutzer* **genau einmal** vor. Die Variable *Transportmittel* besteht dabei nämlich aus dem PKW. Dabei **impliziert** bei klassischen AutoNAVs das (Fehl-)Verhalten des *Benutzers* gleichzeitig das (Fehl-)Verhalten des *Transportmittels*. Der Grund hierfür lässt sich am besten anhand eines Beispiels aufzeigen: Wenn der Benutzer mit einem Auto absichtlich oder unabsichtlich von der vorgegebenen Route abweicht, dann ändert sich dadurch **gleichzeitig** das vorhergesehene Verhalten des *Benutzers* **und** des *Fahrzeugs*. Es kann nicht passieren, dass sich das Verhalten von einer Variable (*Benutzer* oder *Transportmittel*) ändert, **ohne** dass dabei gleichzeitig das Verhalten der anderen Variablen unberührt bleibt.

Anders sieht es bei intermodalen Routen aus. Die Variable „*Transportmittel*“ kommt bei intermodalen Routen **mindestens zweimal** vor (dazu mehr in Kapitel 2). Mit der Zunahme der *Transportmittel* innerhalb der Route steigt dabei gleichzeitig die Anzahl der Variablen. Wie auch bei nicht intermodalen Routen ist jede Variable mit einer Unsicherheit behaftet. Jede einzelne kann dazu beitragen, dass die intermodale Route gefährdet ist. Je mehr *Transportmittel* (also Variablen) somit innerhalb einer Route vorhanden sind, desto größer ist die

Unsicherheit bezüglich des Einhaltens der Route (aus Sicht des *Benutzers*). Zusätzlich kommt hinzu, dass sich die Variablen *Transportmittel* und *Benutzer* **unabhängig** voneinander ändern können (anders als bei nicht intermodalen Routen) – genauso wie die einzelnen Transportmittel ebenfalls. Somit impliziert das (Fehl-)Verhalten einer Variablen **nicht** automatisch das (Fehl-)Verhalten einer anderen Variablen und schon gar nicht das (Fehl-)Verhalten des Benutzers, wie dies bei dem oben beschriebenen Fall eines AutoNAVs ist. Dies ist das **Informationsdefizit**, welches bei intermodalen Routen mit der Anzahl der Transportmittel zunimmt.

Je höher das Informationsdefizit bezüglich der erfolgreichen Absolvierung der intermodalen Route ist, desto **unsicherer** fühlt sich die Person bei der Benutzung einer intermodalen Route. Ab dem Moment, ab dem eine Person zum Beispiel auf der Route in einen Stau gelangt, weiß diese nicht, ob es besser wäre, zum nächsten P&R zu fahren und von dort zum Beispiel mit der S-Bahn weiterzufahren oder doch lieber weiter im Stau stehen zu bleiben. Aktuell würde sich die Person diverser Auskunftsplattformen, wie zum Beispiel der MVV-App⁴ in München oder der DB-App⁵, bedienen um sich über die aktuellen Abfahrtszeiten und die Dauer bis zur Ankunft zu informieren. Das gleiche gilt, wenn ein Benutzer sich bereits in einem öffentlichen Verkehrsmittel befindet und zum Beispiel der Bus, mit dem dieser fahren müsste, eine Verspätung hat. Der Benutzer weiß dann nicht, ob es günstiger wäre, zu warten oder doch lieber zu Fuß zu gehen um dann zum Beispiel mit der U-Bahn (oder einem anderen Transportmittel) weiterzufahren. Es herrscht also innerhalb einer intermodalen Route ein Informationsdefizit, welches die **Handlungsunsicherheit** aus Sicht des Benutzers erhöht.

Informationsdefizit auf einer Route bedeutet, dass jedes einzelne Transportmittel sowie der Benutzer gewissen Störungen unterliegen. Diese Störungen werden in aktuellen Lösungen jedoch nicht miteinander verknüpft und die *Auswirkungen* samt *Alternativen* davon nicht aufgezeigt. Diese Störungen können sowohl aktiver als auch passiver Natur sein (dazu mehr in Kapitel 2). Zudem ist kein Bezug zum Benutzer vorhanden, sodass ein *Echtzeit Guiding*, wie dieses bei einem klassischen AutoNAV existiert, nicht stattfinden kann.

1.2.2 Informationsdefizit - aufgrund fehlender Kommunikation zwischen einzelnen Transportmitteln

Bis auf ein paar wenige Ausnahmen wie beispielsweise *moovel* existieren aktuell sehr wenige intermodale Planer. In den meisten Fällen wird der Benutzer für das Planen seiner Route auf verschiedene Anwendungen zurückgreifen müssen, solche die zum Beispiel die Abfahrtszeiten der U-Bahn mitteilen, oder solche, die den Standort von einem Car-Sharing-Fahrzeug aufzeigen. Das Planen, also in wie weit die Zeiten zusammen passen, um von dem Car-Sharing-Fahrzeug zur U-Bahn zu kommen, bleibt dem Benutzer dabei selbst überlassen. Der große Nachteil ist, dass die einzelnen Systeme voneinander nichts wissen. Sollte es nun innerhalb der intermodalen Route zu einer Störung kommen, sei es bedingt durch den Benutzer oder durch eines der Transportmittel, kennt diese Störung jedoch nur das jeweilige System. In dem beschriebenen Szenario kann es sein, dass die U-Bahn verspätet ist und der Benutzer dies erst erfährt, wenn dieser das Car-Sharing-Fahrzeug wieder freigegeben hat und an der U-Bahn-Station angekommen ist. Im Falle einer Störung stellt sich dem

⁴<http://www.mvv-muenchen.de/>, aufgerufen am 12. Januar 2016

⁵<http://www.bahn.de/>, aufgerufen am 12. Januar 2016

Benutzer nun die Frage, wie dieser jetzt weiterverfahren soll. Soll er auf der bereits gewählten intermodalen Route bleiben oder soll er nach einer Alternative suchen, da er sonst zu spät kommen würde? Dazu müsste der Benutzer wieder sämtliche Insellösungen betätigen, um Alternativen zu finden, und diese bezüglich An- und Abfahrtszeiten sowie Störungen miteinander vergleichen.

1.2.3 Informationsdefizit - aufgrund fehlender Erkennung des aktuellen Transportmittels

Ein intNAV kann nur dann zielgerichtet reagieren, wenn es die tatsächliche Positionierung einer Person kennt. Mit Positionierung ist hierbei jedoch nicht die GPS-genaue Positionierung gemeint. Beispielsweise zeigt *moovel* an, dass laut Fahrplan der Benutzer in drei Minuten an der nächsten Haltestelle aus der S-Bahn Richtung Flughafen aussteigen soll. Dabei weiß *moovel* (und jedes andere aktuelle intNAV) jedoch nicht, ob sich die Person tatsächlich in der S-Bahn Richtung Flughafen befindet. Die Berechnung erfolgt anhand der Planungsdaten, nicht der tatsächlichen Information. Der größte Mehrwert eines intNAVs liegt also darin, innerhalb einzelner Transportmittel *differenzieren* zu können, d.h. zu erkennen ob sich eine Person gerade in einem Auto, einer S-Bahn, einem Bus oder Tram oder einem sonstigen Transportmittel befindet und ob es dabei beispielsweise in der S1, der U3 oder Tram 19 ist, in welche Richtung und welche Haltestelle demnächst angefahren wird. Diese transportmittelgenaue Differenzierung ist aktuell *in der benötigten Reife* nicht vorhanden (vgl. Kapitel 2).

1.2.4 Informationsdefizit - aufgrund fehlender Berechnung der Auswirkungen auf die intermodale Route

Bei einer erkannten Störung innerhalb einer intermodalen Route wäre es für den Benutzer sehr gut zu wissen, inwieweit die Störung seine gesamte intermodale Route betrifft. Führt die Störung womöglich dazu, dass er das nächste Transportmittel nicht erreichen kann, weil die Zeit zu knapp ist, oder ist die Störung so klein, dass der Benutzer trotzdem zum nächsten Transportmittel innerhalb der intermodalen Route kommt. Es kann natürlich sein, dass sich die Störung auf dem aktuellen Transportmittel auch in den anderen wiederfindet. Dies bedeutet, dass die angrenzenden Transportmittel innerhalb der intermodalen Route ebenfalls verspätet sind und somit diese aktuelle Lösung nicht exklusiv ist. Aktuell - bei der Verwendung von Insellösungen - werden Störungen isoliert betrachtet. Es obliegt dem Benutzer eine qualifizierte Aussage darüber zu tätigen, ob die intermodale Route dadurch gefährdet ist oder nicht.

1.2.5 Informationsdefizit - aufgrund fehlender Bewertung der Alternativen

Wenn der Benutzer nun eine Störung bemerkt, dann stellt sich diesem die Frage, wie nun weiter verfahren werden soll. Neben den Auswirkungen auf die intermodale Route, welche weiter oben beschrieben wurden, fehlt bei den aktuellen Systemen eine intelligente Anzeige der Alternativen, welche dem Benutzer nun zur Verfügung stehen. Die Herausforderung dabei ist es zu erkennen, wo die nächste Möglichkeit zu einem Transportmittelwechsel wäre. Es kann beispielsweise sein, dass erkannt wird, dass der Bus, mit welchem der Benutzer aktuell unterwegs ist, zu lange im Stau steht und dadurch die angestrebte Ankunftszeit nicht eingehalten wird. Ein intermodales System, welches nun dem Benutzer mitteilt, er sollte von seinem aktuellen Standort die U-Bahn verwenden, welche gleich in der Nähe liegt, bringt dem Benutzer keinen Mehrwert, da er sich zum aktuellen Zeitpunkt im Bus zwischen zwei Haltestellen befindet und erst bis zur nächsten Haltestelle

warten muss, bevor er aus dem Bus aussteigen kann. Somit nutzt ihm die Information mit der U-Bahn in der Nähe nichts, da die nächste Bus-Station schon zu weit weg ist. Dieses Erkennen, also wann das System nun mit einer Alternativlösung rechnen soll, ist in aktuellen Lösungen ebenfalls nicht vorhanden. Es soll jeweils die Alternativroute mit Bewertung wiedergegeben werden, damit der Benutzer eine richtige Entscheidungsgrundlage hat.

Somit liegt der Kern der vorliegenden Thesis darin, die genannten Informationsdefizite auf intermodalen Routen zu erkennen und zu beheben. In der Thesis sollen dabei die nachfolgend aufgeführten Forschungsfragen beantwortet werden.

FORSCHUNGSFRAGEN UND WISSENSCHAFTLICHER BEITRAG DER VORLIEGENDEN ARBEIT

Damit das beschriebene intNAV funktionieren kann, müssen, wie bereits oben beschrieben, die Informationsdefizite beseitigt werden. Für die vorliegende Thesis lassen sich demnach folgende Forschungsfragen ableiten:

Forschungsfrage 1

„Wie soll die Architektur eines intNAVs aussehen damit die Informationsdefizite beseitigt werden können?“

Ein integriertes intNAV stellt dabei genau das Gegenteil von den aktuellen Insellösungen bei intermodalen Routen dar. Bei einem integrierten intNAV kommunizieren die verschiedenen Planungssysteme der einzelnen Transportmittel miteinander und Störungen werden in der Gesamtheit betrachtet, sodass *Auswirkungen* auf die intermodale Route und *Alternativen* berechnet werden können. Die Architektur eines intNAVs soll nun aufzeigen, wie der Benutzer in Echtzeit auf seiner intermodalen Route geführt werden kann. Die Forschungsfrage 2 schließt sich unmittelbar an Frage 1 an, indem folgendes in Erfahrung gebracht werden sollte:

Forschungsfrage 2

„Wie kann eine Differenzierung der einzelnen Transportmittel, in denen sich die Person aufhält, in Echtzeit erfolgen?“

Dadurch soll erreicht werden, dass das intNAV in Echtzeit erkennt, in welchem Transportmittel sich eine Person befindet, und dieses damit abgleicht, in welchem sich die Person befinden soll. Das ist sinnvoll, da sonst keine richtige Navigation erfolgen kann. Wenn keine Zuordnung stattfindet, kann auch keine Empfehlung gegeben werden, falls es innerhalb des intNAVs zu Störungen kommt. Nachdem nun der Ist-Zustand und der Soll-Zustand bekannt sind, schließt sich folgende Frage an:

Forschungsfrage 3

„Welche Störungen können innerhalb eines intermodalen Navigationssystems auftreten und wie können diese ermittelt werden?“

Um bereits Inhalte aus nachfolgenden Kapiteln vorwegzugreifen, gibt es bei der Unterscheidung von Störungen innerhalb einer intermodalen Route **aktive** und **passive** Störungen. Für die Ermittlung dieser gibt es zum Teil triviale zum Teil aber auch komplizierte Lösungen. Insbesondere bei aktiven Abweichungen gestalten sich der Grad der Störungsermittlung und die Alternativberechnung als nicht trivial.

Nachdem Störungen ermittelt wurden, sollen Auswirkungen auf die Route und die Alternativen errechnet und ausgegeben werden. Daraus resultieren gleich zwei Forschungsfragen, nämlich:

Forschungsfrage 4

„Wie werden Auswirkungen von Störungen auf das intNAV ermittelt?“

und

Forschungsfrage 5

„Wie sollen die Alternativen bewertet werden?“

Der Hintergrund dieser beiden Fragestellungen ist, dass allein der Abgleich zwischen einem Ist- und Sollzustand zwar eine Störung erkennen lässt, jedoch nicht die Frage für ein integriertes intNAV beantwortet, nämlich ob nun eine Umplanung wünschenswert ist oder ob die aktuelle Route weiter verfolgt werden soll. Nur wenn die Auswirkungen einer Störung in die Route mit einberechnet werden, kann daraus geschlossen werden, ob die Störung ein anderes Transportmittel auf der Route betrifft oder nicht.

Nachdem die Störungen samt Auswirkungen und Alternativen errechnet wurden, sollen diese nun dem Benutzer kommuniziert werden. Dabei ist nicht ganz unerheblich zu wissen, *wie* und *wann* ein Benutzer über eine Störung samt deren Auswirkung und Alternativen informiert werden möchte sowie, *was* der Inhalt dieser Information sein sollte. Somit soll die Forschungsfrage 6 diesen Sachverhalt beleuchten:

Forschungsfrage 6

„Wie sollen diese Störungen an den Benutzer weitergegeben werden?“

Zusätzlich stellt sich die Frage, ob die Art und Weise, wie ein Benutzer über eine Störung informiert werden möchte, differenziert werden soll nach der Art der Störung, also ob es sich um eine aktive oder um eine passive Störung handelt. Demnach lautet die nächste Forschungsfrage:

Forschungsfrage 7

„Soll der Benutzer bei einer aktiven Störung anders informiert werden als bei einer passiven?“

Das intNAV verarbeitet in Echtzeit eine sehr große Menge an Daten. Es soll ständig die aktuellste Route auf sämtliche Störungen, welche auftreten können, überprüfen. Gleichzeitig sollen Alternativen ausgegeben werden, welche bewertet werden sollen. Dies alles soll zusätzlich auf einem Smartphone geschehen. Aus diesem Grund schließt sich folgende Frage an:

Forschungsfrage 8

„Existiert eine effiziente Möglichkeit um mit der großen Datenmenge, die auf einem Smartphone verarbeitet werden soll, in Echtzeit umzugehen?“

AUFBAU DER THESIS

Zur Beantwortung der vorgestellten Forschungsfragen wurde die vorliegende Arbeit folgendermaßen aufgebaut:

Nachdem die Motivation für das vorliegende Thema beleuchtet und die Problematik des Informationsdefizits innerhalb eines intNAVs geklärt wurde, sollen im **zweiten Kapitel** die nötigen Grundlagen geschaffen werden, um die vorhandene Forschungslücke rund um das vorliegende Thema aufzudecken. Der Schwerpunkt liegt dabei auf der Identifikation von Möglichkeiten, wie mit Hilfe eines Smartphones das Transportmittel in Echtzeit erkannt werden kann. In diesem Kapitel wird deutlich, dass es das intNAV, wie es in der vorliegenden Arbeit beschrieben wird, noch nicht gibt. Zusätzlich zeigt dieses Kapitel aufgrund der Fülle an aktuell laufenden Arbeiten auf, wie wichtig die vorliegende Forschungsarbeit ist. Ein Patent, welches aus dieser Forschungsarbeit hervorging und 2017 bei BMW zu diesem Thema eingereicht wurde, bestätigt die Wichtigkeit dieses Themas für einen Automobilhersteller. Zusätzlich sollen wichtige Abgrenzungen zu den Themen Mono-, Multi- und Intermodal sowie Routing vs. Guiding gemacht werden.

In Kapitel 3, 4 und 5 wird die im Rahmen der vorliegenden Forschungsarbeit entwickelte Funktionsweise eines intNAVs beschrieben. In **Kapitel 3** wird die Transportmittelerkennung in Echtzeit mittels Smartphonesensoren aufgezeigt. Als einer der wichtigsten Bestandteile eines intNAVs und gleichzeitig einer der Hauptgründe für das vorhandene Informationsdefizit gilt das Fehlen der Transportmittelerkennung in Echtzeit, genauer die Differenzierung der einzelnen Transportmittel, mit welchen der Benutzer in Echtzeit unterwegs ist. Nur wenn klar differenziert werden kann, in welchem Transportmittel sich eine Person in Echtzeit befindet, kann dies mit dem Soll-Zustand abgeglichen werden und auf Änderungen eingegangen werden. Diese Echtzeit Transportmittelerkennung mittels eines Wissensbasierten Systems wird in Kapitel 3 beleuchtet.

Im **Kapitel 4** soll die Architektur eines intNAVs beschrieben werden. Insbesondere soll darauf eingegangen werden, wie die einzelnen Transportmittel miteinander kommunizieren und wie Störungen aufgedeckt werden. Als einer der wichtigsten Punkte dieses Kapitels gilt es zu beleuchten, wie die Auswirkungen von Störungen der einzelnen Modalitäten für die gesamte Route erfasst und ausgewertet werden können.

Da das intNAV mit einer sehr großen Datenmenge arbeitet, wird in **Kapitel 5** beschrieben, wie mit Hilfe des Complex Event Processing (**CEP**) versucht wird, mit dieser umzugehen. CEP wurde ausgewählt, da es ein effizientes Framework darstellt, mit welchem beispielsweise auf dem Aktienmarkt Veränderungen in Echtzeit

registriert werden können. Mit einer Android-Lösung von CEP wurde die vorliegende Architektur umgesetzt. Die Performance samt Benchmark-Tests wird im Kapitel 5 beschrieben.

Damit die Akzeptanz des entwickelten intNAVs bei dem Benutzer verifiziert werden kann, wurden zwei Studien gemacht. Zunächst wurden anhand einer quantitativen Studie mittels eines Fragebogens die Notwendigkeit sowie die Kriterien eines intNAVs verifiziert. Eine zweite, qualitative Benutzerstudie soll die Funktionsweise des im Rahmen der vorliegenden Forschungsarbeit entwickelten intNAV Prototyps testen. Die Ergebnisse der Studien sind in **Kapitel 6** aufgeführt.

Kapitel 7 fasst die Ergebnisse der einzelnen Kapitel zusammen und leitet den jeweiligen wissenschaftlichen Beitrag der vorliegenden Dissertation ab. Das Kapitel schließt mit einem Ausblick auf weitere Arbeiten zu dem Thema intNAV ab.

2 GRUNDLAGEN EINES INTERMODALEN NAVIGATIONSSYSTEMS (INTNAVS)

In diesem Kapitel werden die Grundlagen eines intermodalen Navigationssystems vorgestellt. Dabei wird zunächst auf die einzelnen Elemente eines intNAVs eingegangen und deren Zusammenspiel erläutert (1). Anschließend erfolgt anhand einer Literaturrecherche die Darstellung des aktuellen Standes der Forschung zu den einzelnen Elementen in den Unterkapiteln (2) und (3). Das Ziel dabei ist, die Forschungslücke anhand der existierenden Literatur aufzuzeigen und die Motivation für dieses Thema zu verdeutlichen. Im letzten Abschnitt dieses Kapitels (4) werden die weiteren Schritte der vorliegenden Arbeit beschrieben.

2.1. ELEMENTE EINES INTNAVS

Ein intNAV ist im Gegensatz zu einem traditionellen Navigationssystem, so wie dieses beispielsweise in einem Privat-PKW oder auf einem externen Gerät wie einem Mobiltelefon vorzufinden ist, weniger bekannt. Das intermodale Navigationssystem setzt sich aus den Begriffen „intermodal“ und „Navigationssystem“ zusammen. Beide Begrifflichkeiten sollen nachfolgend beschrieben werden. Aus diesem Grund werden im Unterkapitel 2.1.1 die Begriffe diskutiert. Nachdem die Begrifflichkeiten geklärt sind, wird deutlich, wieso sich die vorliegende Arbeit der Erstellung eines Konzeptes für ein intNAV widmet und nicht beispielsweise dessen Verbesserung. Dies liegt daran, dass es zum Zeitpunkt der Forschung und Publikationen im Rahmen der vorliegenden Arbeit in der Literatur nichts Vergleichbares gab. Ansätze, welche Teile eines intNAVs abdecken, sind veröffentlicht. Jedoch ist das komplette Zusammenspiel eines intNAVs, so wie es im Rahmen der vorliegenden Arbeit erstellt wird, zum Zeitpunkt der Forschung neu. Aus diesem Grund werden nach der Begriffsdefinition, der Aufbau und das Zusammenspiel der einzelnen Elemente des intNAVs aufgezeigt. Dieses so entstandene Konzept bildet den ersten wissenschaftlichen Beitrag der vorliegenden Forschungsarbeit, da es zum Zeitpunkt der Forschung nichts Vergleichbares gegeben hat. Dem Konzept als Beitrag zur Forschung folgt danach die Umsetzung der einzelnen Elemente.

Zwei Elemente des intNAV-Konzepts bilden dabei das Herzstück des erstellten Konzepts, welches im Rahmen der vorliegenden Dissertation entstanden ist: die *Transportmittel-* und *Anomalieerkennung* in Echtzeit. Zu beiden Elementen existieren Publikationen, jedoch weisen alle Publikationen in Bezug auf das intNAV Lücken auf. Dazu wird nachfolgend eine Literaturrecherche in den Unterkapiteln 2.2. und 2.3 den Stand der Forschung

präsentieren. Dabei soll verdeutlicht werden, wo die Forschungslücke liegt, damit ein funktionierendes intNAV erstellt werden kann.

2.1.1 Begriffsdefinitionen

Bei einem intermodalen Navigationssystem stellt sich die Frage, wie der Begriff der Intermodalität abgegrenzt werden kann. In der Literatur sind folgende Differenzierungen zu finden:

MULTIMODALITÄT–INTERMODALITÄT– MONOMODALITÄT

Der Begriff der **Intermodalität** wurde von Chlond und Manz als die „Nutzung unterschiedlicher Verkehrsmittel im Verlauf eines Weges“ definiert (Chlond, et al., 2000). Wohingegen der Begriff der **Multimodalität** definiert wurde als „wechselnde Verkehrsmittelnutzung bei unterschiedlichen Wegen einer Person in einem bestimmten Zeitraum“. Der große Unterschied zwischen der Begrifflichkeit der Multi- und Intermodalität ist laut der Definition der Umsteigevorgang. Wege werden mit Aktivitäten gleichgesetzt. Aktivitäten, welche räumlich voneinander getrennt sind, sind durch Wegeketten miteinander verbunden, z.B. „Fahrt zur Arbeit“ – „Fußweg zur Kantine“ – „Dienstfahrt zum Termin“ – „Fahrt zum Fitnessstudio“ – „Fahrt nach Hause“. Somit kann die

Multimodalität als die *Variation* von Verkehrsmitteln und die Intermodalität als die *Verkettung* von Verkehrsmitteln innerhalb einer Route

gesehen werden. Somit hat der Benutzer bei einem multimodalen Verkehrsverhalten die Möglichkeit verschiedene Verkehrsmittel zu verwenden. Dabei kann jedoch für eine Route ein Verkehrsmittel verwendet werden.

Bei einem Intermodalem Verkehrsverhalten hat der Benutzer die Möglichkeit verschiedene Verkehrsmittel zu verwenden, jedoch nur innerhalb einer Route. Dies bedeutet, dass bei der Intermodalität das Verwenden eines einzigen Verkehrsmittels innerhalb einer Route nicht gestattet wäre. Weiterhin wird die Begrifflichkeit der **Monomodalität** als die *ausschließliche* Verwendung *eines* Verkehrsmittels auf sämtlichen Wegeketten beschrieben. Nachfolgend soll die **Abbildung 3** die Unterscheidung der Begrifflichkeiten „Multimodalität“ und „Intermodalität“ verdeutlichen:

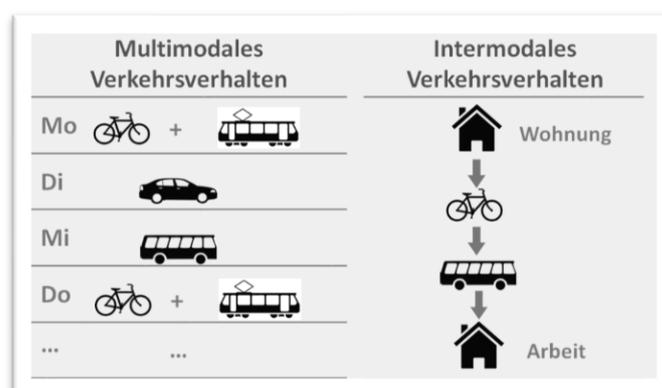


Abbildung 3: Differenzierung der Begriffe Multimodalität und Intermodalität (von der Ruhren, et al., 2003)

Die Begrifflichkeiten können nach von der Ruhren in folgende drei Anwendungsgebiete unterteilt werden (von der Ruhren, et al., 2003):

Beschreibung von Eigenschaften eines Verkehrssystems: *"Dabei beschreibt Multimodalität die grundsätzliche Option für den Nutzer, verschiedene Verkehrsmittel zu verwenden. Intermodalität ergänzt diese Option noch durch die Möglichkeit, zwischen unterschiedlichen Verkehrsmitteln zu wechseln."*

Beschreibung von verkehrspolitischen Strategien: fördern eben diese Eigenschaften des Verkehrssystems (siehe 1.)

Beschreibung des Verkehrsverhaltens von Personen: *"Dabei bezeichnet Multimodalität Verkehrsverhalten, das durch die Verwendung verschiedener Verkehrsmittel im Verlauf eines Zeitraumes, der üblicherweise mehrere Wege beinhaltet, gekennzeichnet ist. Intermodalität ist dagegen definiert als die Nutzung unterschiedlicher Verkehrsmittel im Verlauf eines Weges. Damit ist intermodales Verhalten eine Sonderform multimodalen Verhaltens"*

Somit kann also das klassische AutoNAV als monomodales Navigationssystem gesehen werden, da dem Benutzer ausschließlich für die Navigation das Automobil zur Verfügung steht. Ein Navigationssystem, welches nur Fahrradrouten kennt und das der Benutzer dieses nur für sein Fahrrad verwendet, ist ebenfalls ein monomodales Navigationssystem. Auf der anderen Seite stellt ein Navigationssystem, welches für den Benutzer verschiedene Arten von Verkehrsmittel für eine Route plant, ein intermodales Navigationssystem dar. Zusammenfassend kann also gefolgert werden, dass die Begrifflichkeiten Mono- und Intermodalität Sonderformen der Form Multimodalität darstellen (vgl. **Abbildung 4**).

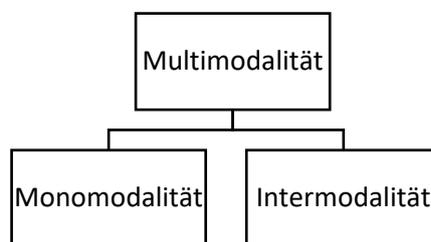


Abbildung 4: Mono- und Intermodalität als Sonderformen der Multimodalität

Durch die Multimodalität stehen dem Benutzer aber auch dem Verkehrssystem mehrere Verkehrsmittel zur Verfügung, mit deren Hilfe eine Route befahren werden kann. Dabei kann diese Route entweder ausschließlich mit einem Verkehrsmittel befahren werden (Monomodalität) oder mit mehreren unterschiedlicher Art (Intermodalität).

GUIDING – ROUTING – NAVIGATION

Nachdem die Differenzierung der Intermodalität geklärt wurde, soll nun der zweite Bestandteil des Begriffs „intermodales Navigationssystem“, also Navigationssystem, geklärt werden. Das Navigationssystem, wie es

beispielsweise aus dem Privat-PKW bekannt ist, kann eine Route *planen* und den Benutzer danach über diese Route *führen*. Die Schritte sehen dabei vereinfacht gesehen so aus:

Der Benutzer gibt in dem Navigationssystem seinen Start- und seinen Zielpunkt ein.

Das Navigationssystem stellt daraufhin dem Benutzer mehrere Routenvorschläge basierend auf der Start-Zieleingabe sowie der Uhrzeit vor.

Der Benutzer wählt basierend auf seinen Bedürfnissen eine der vorgeschlagenen Routen aus.

Das Navigationssystem beginnt den Benutzer auf der ausgewählten Route mit Hilfe seines PKWs in Echtzeit zu führen.

Somit setzt sich das Navigieren aus den Komponenten Planung und Führung zusammen. Der zweite Punkt spricht die **Planung**komponente der Navigation an, wohingegen im vierten Punkt das **Führen** über die Route im Vordergrund steht. **Abbildung 5** stellt den Zusammenhang zwischen den Begriffen Routing und Guiding bildhaft dar und, wie diese zusammen den Begriff Navigation definieren.

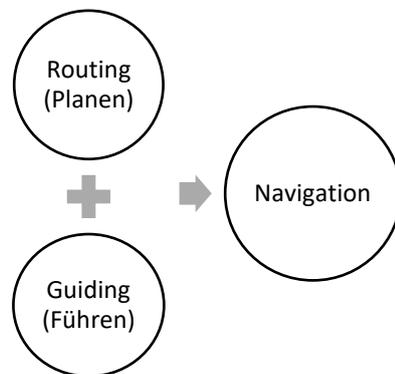


Abbildung 5: Navigation als Summe der Bereiche "Routing" und "Guiding"

Hier setzt zum ersten Mal die große Differenzierung und das Verständnis für die Thematik, mit welcher sich die vorliegende Dissertation auseinandersetzt, ein. Das Routing, also das Planen einer Route, ist weder in der Literatur noch in der kommerziellen Verbreitung sowohl für mono- als auch für intermodale Routen ein neues Forschungsgebiet. Es existieren zahlreiche State-of-the-Art-Lösungen, wie beispielsweise Google Maps, welche dem Benutzer eine Route vom Start- zum Endpunkt planen können – sowohl mono- als auch intermodal. Dabei sind die Routenfindungsalgorithmen mittlerweile so gut, dass eine Forschung in dem Bereich wenig Potential ergibt. Wohingegen das Konzept des Guidings zum Zeitpunkt der Forschung für die vorliegende Arbeit zwar kein Novum im Bereich von monomodalen Navigationssysteme darstellt, jedoch bei intermodalen Routen.

An diesem Punkt setzt nun die vorliegende Dissertation an. In dem Konzept für intNAVs soll also ein Navigationssystem vorgestellt werden, welches in der Lage ist, intermodale Routen zu **planen** und den Benutzer auf diesen zu **führen**. Wie bereits erwähnt, stellt das Planen einer intermodalen Route nicht den Fokus der vorliegenden Arbeit dar, sondern das Führen in Echtzeit über diese Route. Um eine Lösung für dieses Problem zu finden, wurden Elemente in dem Konzept für intNAVs definiert, welche im nächsten Unterkapitel genauer

betrachtet werden. Dabei spielen die Elemente „Transportmittelerkennung“ und „Anomalieerkennung“ in Echtzeit eine zentrale Rolle bei dem intNAV-Konzept und somit in dieser Dissertation.

2.1.2 Aufbau eines intNAVs

Nachdem im vorangegangenen Kapitel die wichtigsten Begriffe geklärt wurden, fällt die Beschreibung des Aufbaus eines intNAVs in diesem Kapitel umso leichter. Damit der Aufbau des intNAVs qualitativ bewertet werden kann, sollen nun nachfolgend Anforderungen an ein intNAV gestellt werden. Dadurch soll gewährleistet werden, dass zunächst die Anforderungen alle sauber definiert sind und somit Verständnisfragen, was nun ein intNAV abdeckt und was nicht, umgangen werden können. Zusätzlich kann am Ende der Dissertation sichergestellt werden, dass die Erfüllung sämtlicher vorgestellter Anforderungen eine erfolgreiche Implementierung eines intNAVs zur Folge hat.

Anforderung 1: Das intNAV soll in der Lage sein, dem Benutzer eine Eingabefläche auf einem Smartphone zur Verfügung zu stellen, bei welcher dieser seinen Start- und seinen Zielpunkt, sowie die gewünschte Uhrzeit eingeben kann. Dabei ist es wichtig, dass die präsentierte Route aus mehreren unterschiedlichen Verkehrsmitteln besteht (=Intermodal). Der Benutzer wählt anschließend eine passende intermodale Route aus.

Anforderung 2: Das intNAV soll in der Lage sein zu erkennen, ob der Benutzer ausgehend von seiner aktuellen Position die Route einhält. Dabei sind sowohl der Routenabschnitt *vor* dem ersten Transportmittel, also der Zeitpunkt wann der Benutzer das Haus verlässt, um zum ersten Transportmittel zu gelangen, als auch die Routenabschnitte *währenddessen* relevant.

Anforderung 3: Das intNAV soll in der Lage sein zu erkennen, ob die gewählten Transportmittel auf der intermodalen Route pünktlich oder abgewichen sind.

Anforderung 4: Das intNAV soll in der Lage sein zu erkennen, ob der Benutzer das richtige Transportmittel im Sinne der zuvor gewählten Route gewählt hat.

Anforderung 5: Das intNAV soll in der Lage sein, bei einer festgestellten Abweichung dem Benutzer die Auswirkungen auf die Route zur Verfügung zu stellen.

Anforderung 6: Das intNAV soll in der Lage sein, bei einer festgestellten Abweichung dem Benutzer die Alternativen zur Verfügung zu stellen.

Bei den dargestellten Anforderungen fällt auf, dass Anforderung 1 und 6 im Bereich des *Routings* anzusiedeln sind und somit, wie bereits erwähnt, nicht Bestandteil dieser Thesis sind. Die Funktionalität von Anforderung 1 und 6 wird zwar in dem hierbei entwickelten Prototypen zur Verfügung gestellt, jedoch ist die Funktionalität dahinter nicht im Rahmen der Thesis entstanden, sondern von einem Fremdsystem bezogen worden. Jedoch sei hier wichtig zu erwähnen, dass in der Anforderung beinhaltet ist, dass das entwickelte intNAV-Konzept nicht an

eine bestimmte Routenberechnungseinheit gebunden ist. Der Fokus der Thesis liegt vielmehr in der Erfüllung der Anforderungen 2 – 5, da sich diese mit dem *Guiding* innerhalb eines intNAVs beschäftigen.

Um also die Anforderungen 2-5 erfüllen zu können, muss das intNAV in der Lage sein, die **Position**

des **Benutzers** zum aktuellen Zeitpunkt t_i auf der Route und
der **Transportmittel** auf der Route zu jedem Zeitpunkt $t_{0...n}$

zu kennen. Wenn die Positionen des Benutzers zum aktuellen Zeitpunkt als auch des Transportmittels zu jedem Zeitpunkt dem intNAV in Echtzeit bekannt sind, muss das intNAV weiterhin in der Lage sein zu bestimmen, ob eine **Abweichung** der

Ist- und Sollposition des **Benutzers** zum aktuellen Zeitpunkt t_i und
Ist- und Sollposition des **Transportmittels** zu jedem Zeitpunkt $t_{0...n}$

auf der Route vorliegt. Diese Schlussfolgerungen von a)-d) stellen eine Verallgemeinerung der Anforderungen 2-5 dar.

Somit lässt sich zusammenfassen, dass folgende zwei Funktionalitäten für das Guiding, sowohl bei mono- als auch intermodalen Routen, von *essentieller* Bedeutung sind: die Positions- und Abweichenbestimmung des Benutzers und der Transportmittel in Echtzeit. Wie auch bei monomodalen Navigationssystemen wird auch bei einem intermodalen eine Erkennung der Abweichung von der Soll-Route erwartet, wenn beispielsweise der Benutzer in das falsche Transportmittel einsteigt oder das relevante Transportmittel sich verspätet hat.

Aus den dargestellten Anforderungen folgt die nachfolgend in **Abbildung 6** dargestellte vereinfachte intNAV-Übersicht. Dabei entsprechen die Zahlen über den Pfeilen in der Abbildung den entsprechenden Anforderungen 1-6. Die grau hinterlegten Kästchen, *Anomalien- und Transportmittelerkennung*, bilden dabei das Herzstück der vorliegenden Arbeit.

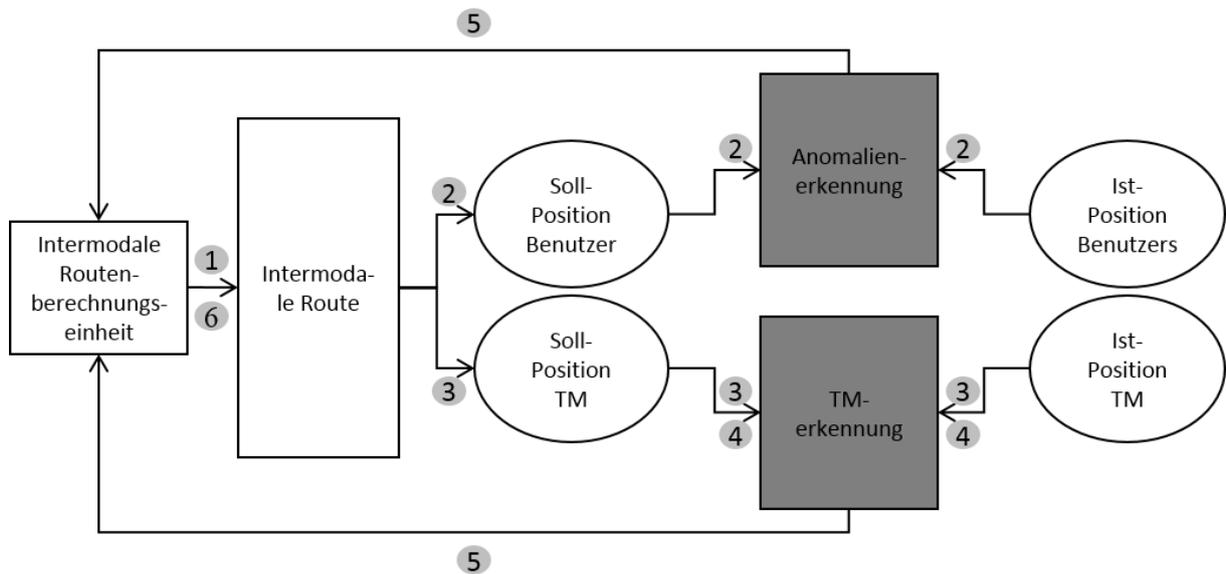


Abbildung 6: Vereinfachte Übersicht der Funktionalität eines intNAVs. Die Zahlen 1-6 entsprechen dabei den Anforderungen 1-6.

Anforderung 1: In der *Intermodalen Routenberechnungseinheit* werden abhängig von dem Start- und Zielpunkt, welche der Benutzer eingegeben hat, intermodale Routen berechnet. Der Benutzer wählt dabei eine passende aus, welche als *intermodale Route* weitergegeben wird. Dabei erhält der Benutzer eine genaue Aufteilung der einzelnen intermodalen Abschnitte, z.B. „Fuß“ → „Auto“ → „Fuß“ → „U-Bahn“ → „Fuß“, mit der entsprechenden Uhrzeit dargestellt. Für die vorliegende Thesis werden diese Komponenten, wie bereits weiter oben beschrieben, extern bezogen. Somit tragen diese beiden Kästchen aus **Abbildung 6** (*Intermodale Routenberechnungseinheit* und *intermodale Route*) nicht als Forschungsergebnis zu dieser Arbeit bei. Im Rahmen der vorliegenden Arbeit wurde eine in-House-Lösung der *BMW Forschung & Technik* verwendet, um intermodale Routen zu berechnen. Jedoch kann eine beliebige intermodale Lösungseinheit für das vorliegende Konzept verwendet werden. In **Kapitel 3** und **5** wird auf die autarke Kopplung beliebiger Routenberechnungseinheiten eingegangen. Wichtig hierbei ist nur, dass das intNAV die Verkehrsmittel kennt, aus denen die intermodale Route bestehen soll. Beispielsweise wird das vorliegende intNAV-Konzept nichts mit dem Verkehrsmittel „Transrapid“ anfangen können, da dies im Rahmen der vorliegenden Arbeit mangels Testzwecke ausgeklammert wurde.

Anforderung 2: Um nun Abweichungen von der gewählten Route festzustellen, bezieht die Komponente *Anomalien-erkennung* die Position des Benutzers zu einem bestimmten Zeitpunkt t_i von der *Intermodalen Routenberechnungseinheit*. Dadurch erhält die Komponente die Soll-Position des Benutzers, also wo sich der Benutzer laut Planungseinheit, zum Zeitpunkt t_i befinden sollte. Zusätzlich bezieht die *Anomalien-erkennungs-*Komponente eine weitere Position des Benutzers, nämlich die tatsächliche Position zum gleichen Zeitpunkt t_i . Die *Anomalien-erkennungs-*Komponente vergleicht dann die beiden Positionen zum Zeitpunkt t_i . Wenn eine Abweichung, also eine Anomalie, festgestellt wurde, erkennt diese Komponente die Abweichung, woraufhin Anforderung 2 erfüllt wäre. In **Kapitel 4** wird die Funktionsweise der Anomalien-erkennungs-komponente beschrieben.

Anforderung 3: Neben der Anomalieerkennung ist für das Guiding innerhalb eines intNAVs die *Transportmittelerkennung* in Echtzeit relevant. Die Komponente *Transportmittelerkennung* bezieht dabei, ähnlich wie die *Anomalieerkennung*, zwei Positionen des Transportmittels zum Zeitpunkt t_i : die Soll- und die Ist-Position eines Transportmittels, wie in **Abbildung 6** zu sehen ist. Wird dabei eine Abweichung zwischen den beiden Positionen zum Zeitpunkt t_i festgestellt, ist die Anforderung 3 erfüllt. Die Funktionsweise der Transportmittelerkennung ist ein wesentlicher Bestandteil des **Kapitels 3** und wird auch im **Kapitel 5** behandelt.

Anforderung 4: Die *Transportmittelerkennungs*-Komponente soll jedoch nicht nur die Abweichung des Transportmittels zum Zeitpunkt t_i erkennen, sondern auch, ob sich der Benutzer zum Zeitpunkt t_i in dem korrekten Transportmittel befindet. Dadurch soll erkannt werden, wenn der Benutzer (un)beabsichtigt in das falsche Transportmittel eingestiegen ist. Natürlich kann der Benutzer aber auch absichtlich in ein anderes Transportmittel, als es ihm von der *Routenberechnungseinheit* vorgegeben wurde, eingestiegen sein. Die Gründe dafür könnten zum Beispiel sein, dass dem Benutzer noch eine Erledigung eingefallen ist, welche dieser noch machen kann, da der Termin unkritisch ist. Wenn das intNAV nun diese gewollte oder ungewollte Abweichung erkennt, gilt **Anforderung 4** als erfüllt. In **Kapitel 3** wird diese Transportmittelerkennung aus Sicht des Benutzers eingehend beleuchtet.

Anforderung 5: Wenn nun eine oder beide Komponenten eine Abweichung registrieren, soll das intNAV dies dem Benutzer adäquat mitteilen. Was unter „adäquat“ zu verstehen ist, wird in **Kapitel 6** untersucht. Dabei wurde neben einer Umfrage auch eine Studie durchgeführt, wie die Art und Weise einer erkannten Abweichung dem Benutzer auf dem Smartphone zu präsentieren ist. Die Umsetzung der Anforderung 5 stellt lediglich einen Output der beiden Komponenten *Anomalien- und Transportmittelerkennung* dar und gleichzeitig den Input für die intermodale Routenberechnungseinheit. Dies bedeutet, dass im Rahmen der vorliegenden Arbeit bei einer erkannten Abweichung dem Benutzer zwar Alternativen vorgestellt werden, diese jedoch von einer externen Quelle bezogen werden und nicht zusätzlich interpretiert werden (siehe letzte Anforderung, **Anforderung 6**). Der wissenschaftliche Beitrag im Rahmen dieser Anforderung liegt darin, dass eine Untersuchung stattgefunden hat, wie der Benutzer Abweichungen in intermodalen Routen kommuniziert haben möchte und ob dies zusätzlich von der Art der Abweichung und dem Kontext, in welchem sich der Benutzer befindet, abhängt.

Anforderung 6: Nachdem eine Abweichung/Anomalie von der Soll-Route festgestellt wurde, sollen dem Benutzer Alternativen zu seiner gewählten Route präsentiert werden. Dabei handelt es sich um eine Neu-Berechnung der Route, ausgehend von dem aktuellen Startpunkt.

Wie in der **Abbildung 6** dargestellt und auch im Fließtext dazu beschrieben wurde, stellen die Komponenten *Anomalieerkennung* und *Transportmittelerkennung* den wesentlichen Mehrwert der vorliegenden Arbeit dar. Dies liegt daran, dass diese beiden Komponenten das **Guiding** in einem Navigationssystem abbilden, wohingegen die Komponente intermodale Routenberechnungseinheit das **Routing** im Navigationssystem nachbildet (vgl. **Abbildung 5**). Da das Guiding, also die beiden Komponenten *Anomalieerkennung* und *Transportmittelerkennung*, im Gegensatz zum Routing, in intNAVs noch nicht erforscht ist, bildet es den Hauptteil der vorliegenden Forschungsarbeit.

Somit lässt sich die in **Abbildung 5** dargestellte Zusammensetzung des Navigationssystems durch die beiden Bereiche *Guiding* und *Routing* auf das intNAV, wie in **Abbildung 7** dargestellt, anpassen:

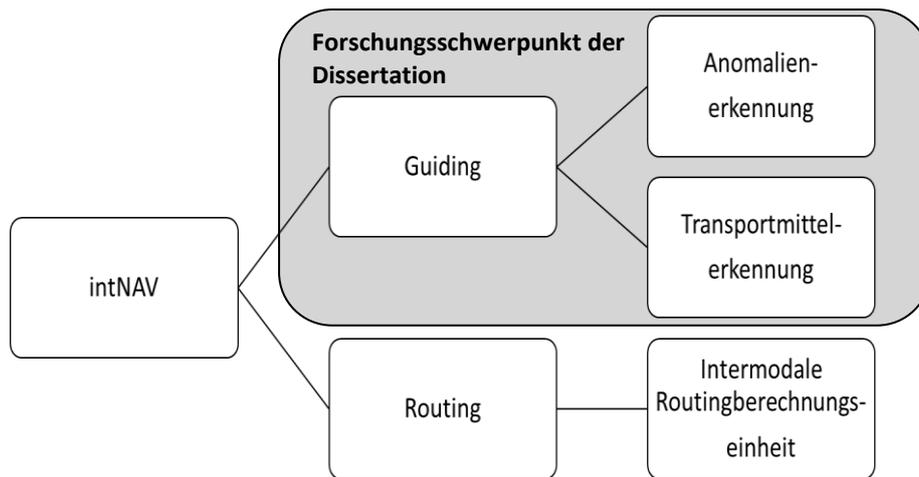


Abbildung 7: Bestandteile eines intNAVs und Einordnung des Forschungsschwerpunktes der Doktorarbeit.

Um nun die Forschungslücke zum Thema Guiding in intNAVs aufzuzeigen, wird in dem nachfolgenden Unterkapitel 2.2 eine Literaturrechercheen Stand der Technik wiedergeben. Anschließend folgt in den Kapiteln 3 und 4 die konkrete Umsetzung des Guidings in intNAVs.

2.2. TRANSPORTMITTELERKENNUNG MITTELS SMARTPHONE-SENSOREN IN ECHTZEIT

Wie bereits erwähnt, stellt die Transportmittelerkennung einen wichtigen Bestandteil des intNAVs dar. Dabei soll das intNAV, wie es in **Anforderung 1** unter Kapitel 2.1.2 beschrieben steht, auf einem Smartphone ausführbar sein. Der Vorteil eines Smartphones liegt darin, dass sich dieses als ein ständiger Begleiter des Menschen etabliert hat (Markowetz, 2015). Aus diesem Grund besteht die Herausforderung darin, die verfügbaren Smartphone-Sensoren hinsichtlich deren Eignung für die Transportmittelerkennung zu beurteilen und die vielversprechendsten darunter für die Transportmittelerkennung zu verwenden.

Bestehende Forschungsarbeiten verfolgen dabei den Ansatz, entweder ausschließlich einzelne Smartphone-Sensoren oder eine Kombination von verschiedenen Sensoren mit zusätzlicher Kontextinformation zu verwenden, um eine gute Transportmittelerkennung zu gewährleisten. Bei der Literaturrecherche wurde schnell deutlich, dass die Verwendung einzelner oder einer Kombination mehrerer Sensoren für eine Transportmittelerkennung nur mäßig gut geeignet sind. In Reddy et al. wird beschrieben, dass bei der Verwendung einzelner Sensoren gegenüber der Verwendung mehrerer Sensoren mit einer Einbuße von 10-20% hinsichtlich der Erkennung eines Transportmittels auszugehen ist (Reddy, et al., 2010). Zwar lassen sich dadurch *motorisierte* von *nicht-motorisierten* Zuständen unterscheiden, jedoch gestaltet sich die Unterscheidung innerhalb motorisierter Zustände, z.B. *Auto, Bus, S-Bahn*, etc. als deutlich fehleranfälliger. Jedoch wird gerade diese Granularität der Differenzierung innerhalb motorisierter Zustände für die vorliegende Arbeit benötigt.

Aus diesem Grund wird in der Literatur zusätzlich zur Verwendung von Sensoren auch Kontextinformation hinzugenommen, um beispielsweise nicht nur eine Differenzierung zwischen motorisiert und nicht-motorisiert vorzunehmen, sondern auch eine Unterscheidung zwischen den einzelnen motorisierten Zuständen zu ermöglichen und dabei folgende Fragestellungen beantworten zu können: „Welche Route fährt der Bus, in dem sich die Person aktuell befindet?“ oder „Wie lautet die nächste Station, welche die Tram ansteuert, in welcher sich der Benutzer aktuell befindet?“ oder auch „Wo befinden sich die Ein- und Ausgänge zu der S-Bahn-Station der S-Bahn-Linie, welche der Benutzer demnächst benötigt?“. Für die vorliegende Dissertation wird genau diese Informationstiefe in einem intNAV benötigt und somit liegt die Verwendung von mehreren Sensoren samt Kontextinformation für die Transportmittelerkennung nahe.

Dabei werden in der Literatur zwei Stränge verfolgt um aus diesen Sensoren ein „komplexes“ Verhaltensmuster, wie z.B. *gehen, laufen, Treppen steigen, Auto oder Bus fahren* abzuleiten: **Maschinelles Lernen** oder **Regelbasierte Systeme** (Partsch, et al., 2012) (Patterson, et al., 2003) (Reddy, et al., 2008) (Reddy, et al., 2010). Obwohl Regelbasierte Mechanismen einfacher zu verstehen und anzuwenden sind als Methoden des Maschinellen Lernens, sind diese schwierig zu generalisieren. Die Regeln, welche innerhalb eines Algorithmus angewendet werden, können zwar sehr gute Resultate für ein bestimmtes Problem liefern, sind jedoch sehr schwierig auf andere Probleme anzuwenden. Am offensichtlichsten ist es, wenn Regeln für die Transportmittelerkennung innerhalb einer bestimmten Stadt aufgestellt werden. Diese auf eine andere Stadt zu übertragen, gestaltet sich als sehr schwierig.

Nachfolgend werden relevante Arbeiten basierend auf einer **Sensorfusion**, also der Verwendung von mindestens zwei Smartphone-Sensoren, analysiert und deren Erkennungswahrscheinlichkeit bezüglich der Transportmittel gegenübergestellt. In den meisten Fällen wurden dabei Kombinationen aus dem **GPS**-Empfänger und anderen Sensoren verwendet. Letztendlich findet sich nachfolgend eine Literaturrecherche, bei welcher zusätzlich zu den Sensoren auch **Kontextinformation** hinzugenommen wurde wie beispielsweise das zu Grunde liegende öffentliche Verkehrsnetz samt eines Echtzeit-Fahrplans der Verkehrsmittel. Eine zusammenfassende Tabelle am Ende dieses Kapitels gibt einen Überblick über die *verwendete Methodik, erkannten Transportmittel, Erkennungswahrscheinlichkeit* sowie, ob diese Erkennung in *Echtzeit* funktioniert hat oder nicht. Für das intNAV ist die Echtzeiterkennung unabdingbar.

Das Ergebnis der Literaturrecherche ist nachfolgend dargestellt. Dabei werden zunächst die relevanten Sensoren und die entsprechende Güte der Transportmittelerkennung beleuchtet.

2.2.1 Transportmittelerkennung mittels einzelner Sensoren

Moderne Smartphones besitzen eine Vielzahl verschiedener Sensoren für jeweils unterschiedliche Anwendungsgebiete. Beispielsweise kann mit Hilfe des Barometers der Luftdruck auf bis zu 0,12 hPa genau bestimmt werden (Bosch, 2017). Dabei handelt es sich um einen Luftdrucksensor von Bosch, welcher am häufigsten in Smartphones verbaut wird. Dies entspricht ca. einem Meter Höhenunterschied, wodurch sich beispielsweise Treppensteigen mit Hilfe des Barometers feststellen lassen könnte. Die Sensoren moderner Geräte sind so gut ausgeprägt, dass eine sehr genaue Messung in jeweils entsprechenden Anwendungsfeldern

erfolgen kann. Von einem Helligkeitssensor über einen W-LAN-, Bluetooth- und GPS-Empfänger, einem Thermometer- und Feuchtigkeitssensor, einem Barometer, Näherungssensor, einen Rotationssensor, einen Beschleunigungssensor bis hin zu einem Magnetometer verfügen moderne Geräte über eine Reihe verschiedenster Sensoren (Macias, et al., 2013).

Nicht alle der erwähnten Sensoren sind jedoch gleich gut geeignet, um ein Transportmittel in Echtzeit zu identifizieren. Zur Bestimmung eines Transportmittels haben sich laut der durchgeführten Literaturrecherche bestimmte Sensoren bewährt. Dazu gehören der **GPS-Sensor**, der **Beschleunigungs-** und **Rotationssensor**, das **Gyroskop**, das **Barometer** sowie in jüngster Zeit auch das **Magnetometer**.

Transportmittelerkennung ausschließlich anhand des GPS-Sensors

Zur Positionsbestimmung mittels Global Positioning System (**GPS**) werden die Signale von drei Satelliten von dem GPS-Sensor am Smartphone empfangen und die Position des Gerätes und somit des Benutzers mittels Triangulation bestimmt. In den 1990er Jahren wurde die Verwendung von GPS in sogenannten Fahrgastbefragungen für die statistische Erhebung der Fahrgastinformation immer beliebter (Xiao, et al., 2015) (Yue, et al., 2014) (Murakami, et al., 1997). Dabei sollten unter anderem der Grund für die Reise, Start und Ziel sowie die Fortbewegungsart dokumentiert werden. Mittels GPS konnten dabei *einfache Informationen* wie Zeitstempel oder die Position bestimmt werden und mussten nicht manuell erhoben werden. Diese Automatisierung hatte eine niedrigere Fehleranfälligkeit zur Folge. Daraufhin wurde nach Möglichkeiten gesucht, wie mittels GPS-Daten *komplexe* Informationen über eine Reise gewonnen werden können (Gong, et al., 2014).

Ein Beispiel hierzu liefert die Forschergruppe rund um *Gonzalez et al.* (Gonzalez, et al., 2009), die neben dem reinen GPS-Log zusätzlich Neuronale Netze zur Transportmittelerkennung verwenden. Bei dem dabei resultierenden System TRACT-IT wurden mittels 10-facher Kreuzvalidierung die GPS-Daten trainiert und getestet. Bei diesem Vorgehen wurde eine Erkennungsquote von 92,11% für das *Auto*, 81,58% für den *Bus* sowie 100% für den Modus *zu Fuß* erreicht. Dieser Ansatz unterliegt jedoch Einschränkungen. Zum einen muss der Benutzer stets seinen Start- und Endpunkt manuell eintragen. Viel wichtiger ist dabei jedoch, dass das System nicht über eine *automatische* Segmentierung der einzelnen Wegstrecken verfügt. Für ein intNAV ist diese Lösung somit nicht praktikabel, da in Echtzeit die Segmentierung und die automatisierte Transportmittelerkennung erfolgen müssen. Zusätzlich wurde nur die Transportmittelerkennung zwischen drei Modi durchgeführt, genauer genommen nur zwei Modi, wenn der Benutzer sich innerhalb der motorisierten Zustände bewegt. Wie jedoch bereits erwähnt, muss ein intNAV je nach Region in der Lage sein, mehrere verschiedene Transportmittel zu identifizieren, welche in einem bestimmten Ort zu dem ÖPNV gehören. Ebenfalls mittels Neuronaler Netze gelang es dem Forscherteam rund um *Yang et al.* (Yang, et al., 2015) eine Erkennungsrate für den Transportmodus *Bus* von 86% zu erzielen. *Xiao et al.* (Xiao, et al., 2015) verwendeten für die Transportmittelerkennung (**TME**) eine Kombination aus Neuronalen Netzen (**NN**) sowie der Partikelschwarmoptimierung (**PSO**) die sie auf die erhobene GPS-Koordination angewendet haben, um zwischen den Modi *gehen, Fahrrad, Bus und Auto* zu unterscheiden. Dabei wurde eine Erfolgsquote von maximal 94,44% erzielt. Auch hier fällt auf, dass für ein intNAV dieser Ansatz nicht geeignet ist, da weitere öffentliche Verkehrsmittel außen vor gelassen wurden.

Für ihre Fahrgaststudie wurden im Zeitraum von Mitte Oktober 2013 bis Mitte Juli 2014 von *Xiao et al.* (Xiao, et al., 2015) GPS-Koordinaten aufgezeichnet. Mit diesem Datensatz wurde ein Bayes'sches Netz generiert, welches die Transportmittel *gehen, Fahrrad, E-Fahrrad, Auto und Bus* identifizieren sollte. Die Erkennungswahrscheinlichkeit lag bei 92,74%, wobei festgestellt werden konnte, dass durch die Anwendung des Bayes'schen Netzes eine eindeutigere Differenzierung zwischen den Modi *Auto* und *Bus* möglich ist. Der Nachteil dieser Forschungsarbeit ist jedoch, dass die Erkennung nicht in Echtzeit funktioniert.

Gegenüber den anderen Arbeiten, konnten *Zhang et al.* (Zhang, et al., 2011) durch die Verwendung von Support Vektor Maschinen (**SVM**) neben den bekannten Modi *gehen, Fahrrad, Bus und Auto* auch die Modi *Zug* und *Tram* identifizieren. Dabei ist auffällig, dass die Erkennungswahrscheinlichkeit der Modi *Auto* und *Zug* bei 100% liegt, die der *Tram* jedoch nur bei 78%.

Die Forschergruppe um *Gonzalez et al.* (Gonzalez, et al., 2009) aus dem Jahr 2009 untersuchte verschiedene GPS-Tracks und stellte dabei fest, dass sich zwischen den Tracks von Bussen und Autos fast keine Unterschiede finden ließen, wohingegen die Tracks von Fußgängern sich leicht von den motorisierten GPS-Aufzeichnungen unterscheiden lassen. In **Abbildung 8** wird der Unterschied zwischen den aufgezeichneten GPS-Logs eines Fußgängers und eines Autofahrers (resp. Busfahrers) klar. Zwischen den GPS-Tracks eines Busses sowie eines Autos findet man jedoch kaum Unterschiede.

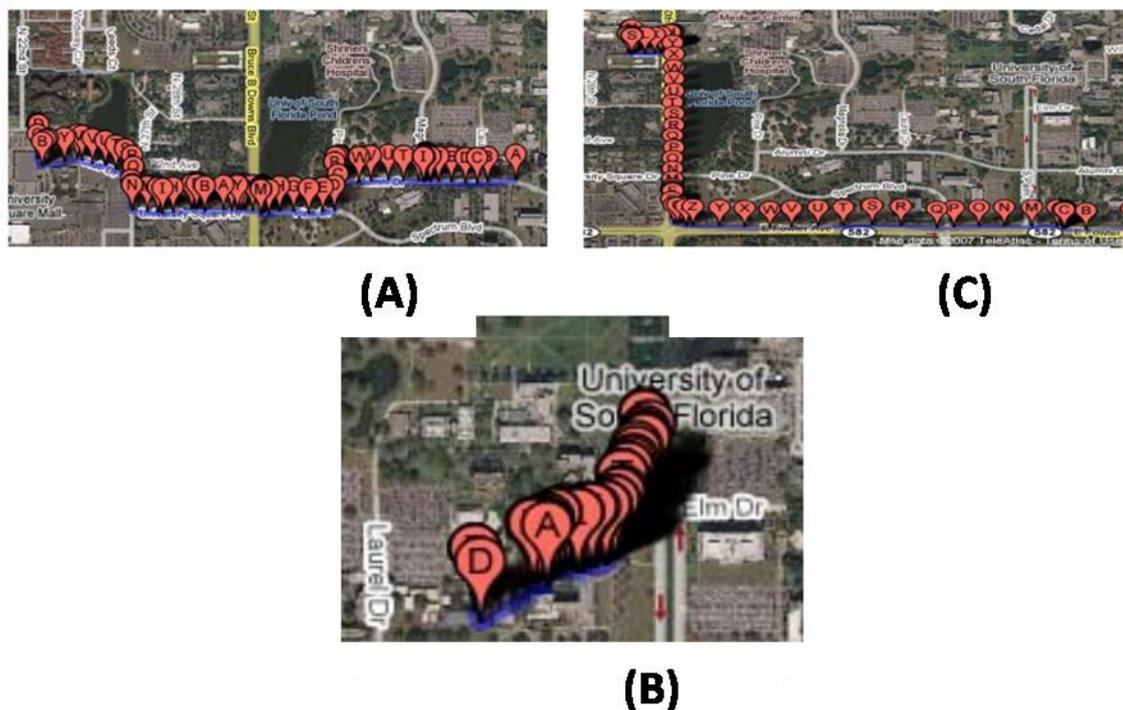


Abbildung 8: GPS-Logs der Fortbewegungsarten Bus (A), zu Fuß (B) und Auto (C) (Gonzalez, et al., 2009)

Zu einem interessanten Ergebnis kamen *Reddy et al.* (Reddy, et al., 2010), welche den Einfluss der Smartphone-Position auf die Erkennungswahrscheinlichkeit der Transportmittel untersuchten. Im ersten Schritt wurde ein Decision Tree (**DT**) Algorithmus angewendet um im zweiten Schritt ein Hidden Markov Model (**HMM**)

anzuwenden. Damit wurde eine Erfolgsquote von 93,6% erzielt. Dabei wurden 16 Teilnehmer gebeten, 15 Minuten Daten für jedes der Transportmittel, *gehen*, *Fahrrad*, *Auto* zu sammeln, dabei jedoch das Smartphone in verschiedenen Positionen zu haben. Dabei wurde festgestellt, dass die Positionierung (am Arm, an der Tasche, in der Hand, in der Hosentasche oder am Becken) keine signifikanten Auswirkungen auf die Erkennungswahrscheinlichkeit hat, wobei die beste bei der Position „Hand“ (95%) und die schlechteste „in der Tasche“ (94,3%) gemessen wurde.

In einer weiteren Untersuchung, welche an 65 Probanden 10 Monate lang durchgeführt wurde, konnte mit einer Erkennungsrate von 78% eine Unterscheidung zwischen den Modi *gehen*, *Fahrrad* und *Auto* durch den Einsatz des GPS-Sensors am Smartphone erreicht werden (Zheng, et al., 2008).

Zusammenfassend kann also gefolgert werden, dass sich für die Transportmittelbestimmung im intNAV die alleinige Verwendung von GPS als wenig zuverlässig erwiesen hat. Das GPS-Signal kann durch schlechtes Wetter oder Störsignale wie beispielsweise Gebäude beeinträchtigt werden. Die untersuchten Arbeiten zeigen deutlich, dass die Verwendung von GPS sehr sinnvoll ist, um eine Unterscheidung zwischen motorisierten und nicht-motorisierten Zuständen zu ermöglichen, es sich jedoch als schwierig erweist, alleine mit Hilfe des GPS-Signals eine Unterscheidung zwischen den motorisierten Zuständen zu erhalten. Aus diesem Grund widmen sich zahlreiche Publikationen einer hybriden Lösung, bei welcher neben den GPS-Koordinaten noch weitere Sensoren hinzugenommen werden um durch logische Schlussfolgerungen den Transportmodus zu bestimmen.

Transportmittelerkennung ausschließlich anhand des Beschleunigungssensors

Ein weiterer Sensor mit dessen Hilfe erkannt werden kann, welche Fortbewegungsart gerade vorliegt, ist der Beschleunigungssensor (Accelerometer). Das Smartphone kann sich in drei Richtungen $\langle x, y, z \rangle$ ausrichten (vgl. **Abbildung 9**). Der Accelerometer misst die Beschleunigung jeweils in die drei Richtungen und liefert dies als 3D Beschleunigung zurück. Im Gegensatz zum GPS-Sensor ist der Beschleunigungssensor weniger störanfällig aufgrund von äußeren Einflüssen.

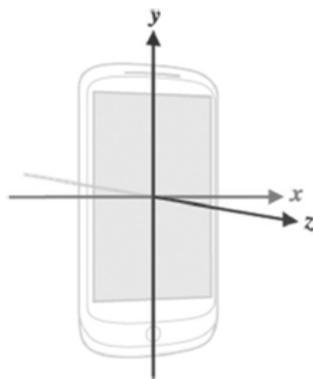


Abbildung 9: $\langle x, y, z \rangle$ -Achsen eines Smartphones

Abbildung 10 zeigt die Transportmittelerkennung basierend auf der reinen Accelerometer-Messung für die Fortbewegungsarten *Fuß* und *Bus* (Su, et al., 2014). Dabei konnten mittels des Beschleunigungssensors starke

Ausschläge für die Fortbewegungsart *zu Fuß* nachgewiesen werden, wohingegen bei dem Transportmittel *Bus* kaum Ausschläge des Beschleunigungssensors wahrzunehmen sind. Auch in der Arbeit von *Siirtola und Röning* wird eine Transportmittelerkennung mit Hilfe eines Drei-Achsen Accelerometer im Smartphone durchgeführt und auch dort zeigt sich ein klares Ergebnis (Siirtola, et al., 2012). Die Modi welche dabei in Echtzeit festgestellt werden konnten sind *zu Fuß*, *Fahrrad*, *Auto*, jedoch nicht die Unterscheidung zwischen *Auto* und *Bus* oder *S-Bahn*. Die Transportmittelerkennung wurde dabei mittels der Algorithmen Quadratische Diskriminanten Analyse (**QDA**) und k-nächster Nachbar (**KNN**) untersucht und verglichen. Dabei fällt auf, dass der Modus *Auto* mit einer sehr schwachen Wahrscheinlichkeit von 67,8% korrekt erkannt wird. *Gehen* und *Fahrrad* hingegen konnten mittels eines reinen Accelerometers auf einem Smartphone mit einer Wahrscheinlichkeit von 96,5% korrekt erkannt werden.

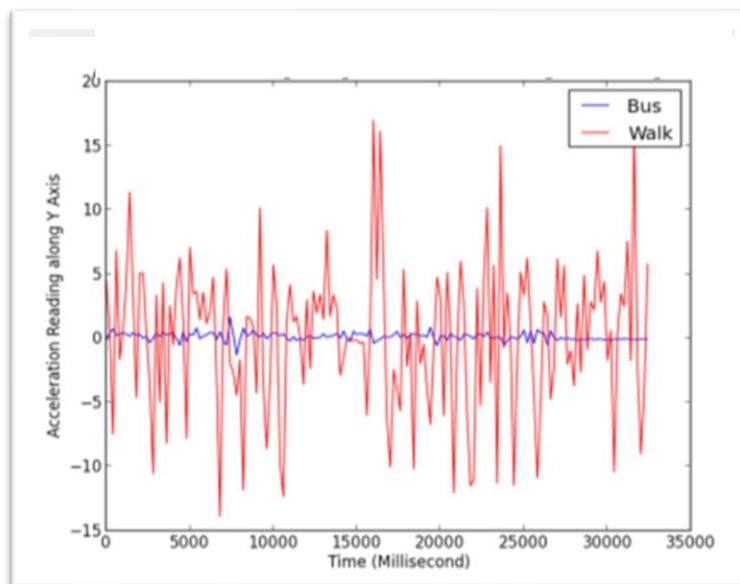


Abbildung 10: Beschleunigungserkennung für die Zustände Gehen (rot) und Bus (blau) (Su, et al., 2014)

Zusammenfassend lässt sich festhalten, dass die Transportmittelerkennung basierend auf der ausschließlichen Verwendung des Beschleunigungssensors gute Resultate liefert, um eine Generalisierung zwischen den Zuständen „motorisiert“ und „nicht-motorisiert“ zu erhalten, jedoch versagt, wenn eine Unterscheidung innerhalb von motorisierten Zuständen, wie *Auto*, *Bus* oder *Tram* benötigt wird. Im Gegensatz zum GPS-Sensor ist sie zudem weniger Fehleranfällig durch schlechtes Wetter oder Umgebungsparameter. Diese Unterscheidung innerhalb der motorisierten Zustände ist jedoch für die Transportmittelerkennung innerhalb eines intNAVs, bei welchem intermodale Routen mit Bussen, S-Bahnen, Trams etc. befahren werden, essentiell. Die alleinige Verwendung des Beschleunigungssensors scheidet auch aus dem Grund aus, da sich damit nicht die Position und die Ausrichtung des Benutzers feststellen lassen.

Transportmittelerkennung ausschließlich anhand des Barometers

Mit Hilfe des Drucksensors (Barometers) kann der Luftdruck gemessen werden. In einer sehr dünnen (zehn Mikrometer dicken) Siliziummembran sind Dehnungstreifen eingelassen. Die Dehnungstreifen können ihren Widerstand ändern. Die Membran bewegt sich bei Druckveränderungen, wodurch sich die Widerstände verändern. Das gesamte Modul des gängigsten Sensors von Bosch ist dabei nur zwei mal zweieinhalb Millimeter dick (Bosch, 2017).

Muralidharan et al. zeigten in einem Experiment, dass sich dieser abhängig vom Gebäudetyp und der –struktur verändert und dass dieses Verhalten gelernt werden kann (Muralidharan, et al., 2014). Die Forschergruppe um *Su et al.* zeigte in einem Experiment, dass mit Hilfe des Barometers das Transportmittel U-Bahn sehr gut identifiziert werden konnte (Su, et al., 2014). Ihre Vermutung dabei ist, dass der Luftdruck unterirdisch stabiler ist als oberirdisch und somit unterirdisch fahrende Transportmittel mit Hilfe des Barometers besser erkannt werden können als beispielsweise ein *Bus* oder ein *Auto*. In **Abbildung 11** sind die jeweiligen Aufzeichnungen der einzelnen Barometerverläufe der Transportmittel *Bus*, *Auto* und *U-Bahn* dargestellt. Auch hier wird deutlich, dass sich die Unterscheidung der einzelnen motorisierten Zustände bis auf das Auto als eine Herausforderung erweist und somit scheidet auch die alleinige Verwendung des Barometers zur Transportmittelerkennung im intNAV aus.

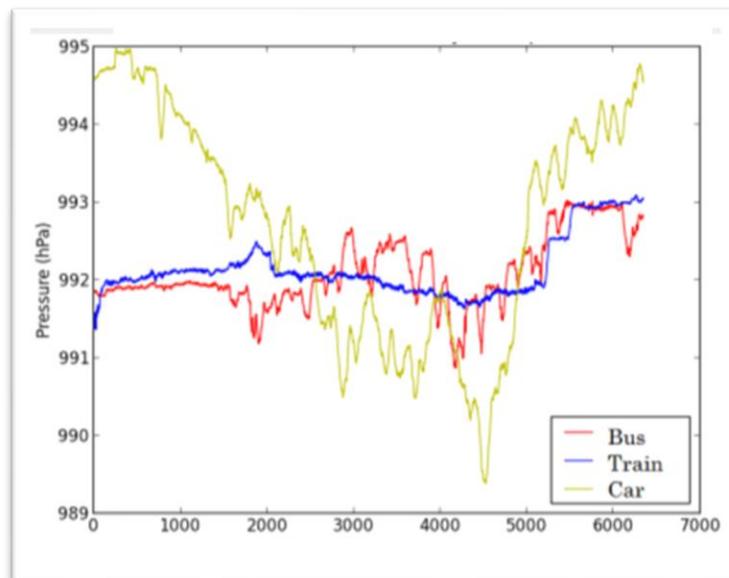


Abbildung 11: Unterschiede in der Luftdruckerkennung bei den Transportmodi Auto (gelb), Bus (rot), U-Bahn (blau) (Su, et al., 2014)

Transportmittelerkennung ausschließlich anhand des Magnetometers

Der Magnetometer ist als ein vier mal vier Millimeter großer Chip in einem Smartphone zu finden, welcher die erdmagnetische Anziehung an den drei Achsen $\langle x,y,z \rangle$ vermisst. Das Erdmagnetfeld hat je nach Ort auf dem Planeten eine Flussdichte von 30 bis 60 Mikrottesla. Der Sensor in einem iPhone 4 kann Veränderungen von 0,3 Mikrottesla registrieren, was genügt um Stromleitungen in Wänden zu erkennen. In einem Experiment, bei

welchem die Modi zu *Fuß*, *Bus* und *U-Bahn* untersucht wurden, konnte nachgewiesen werden, dass der Ausschlag beim magnetischen Feld bei dem Modus *U-Bahn* am höchsten war.

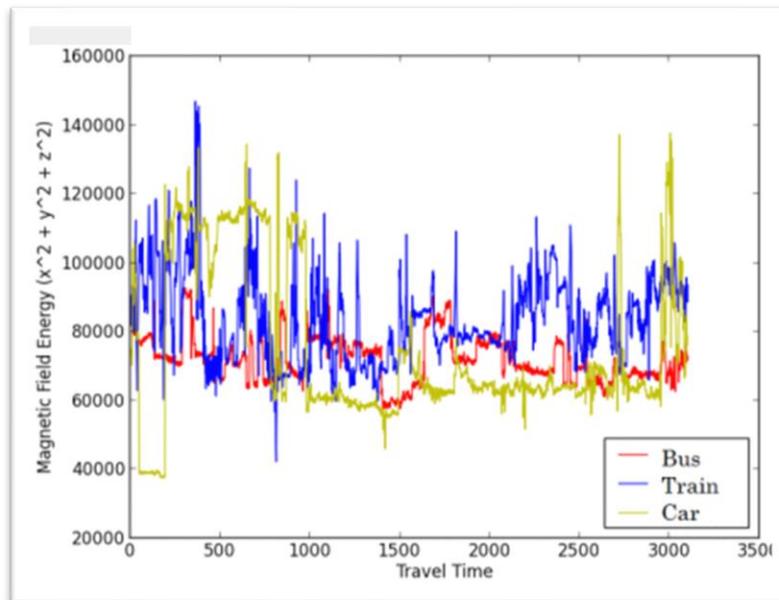


Abbildung 12: Aufzeichnungen der Transportmodi mittels magnetischem Feld (Su, et al., 2014)

Die Vermutung der Forscher lautet daher, dass dies aufgrund des Zugmaterials als auch aufgrund der Elektrizität ausgelöst wird (vgl. **Abbildung 12**). Auch eine Forschungsarbeit bei der BMW Forschung & Technik hat sich diesem Thema gewidmet und festgestellt, dass die Anfahrts- und Abfahrtskurve von U-Bahnen in München mit Hilfe des Magnetometers sehr genau aufgezeichnet werden kann.

Wie auch der **Abbildung 12** entnommen werden kann, lässt sich jedoch das Transportmittel *Bus* relativ schwer mit Hilfe des Magnetometers unterscheiden, wodurch es damit als alleiniger Sensor für die Transportmittelerkennung ausscheidet.

Die Literaturrecherche über die ausschließliche Verwendung der gängigsten Sensoren zur Transportmittelerkennung lieferte dabei folgende Ergebnisse: Die Transportmittel, welche in den Veröffentlichungen am häufigsten untersucht wurden, sind zu *Fuß*, *Auto*, *Bus* und *Zug* (wobei unter *Zug* in manchen Fällen *S-Bahnen* verstanden werden kann). Ein Ergebnis der Literaturrecherche ist weiterhin, dass eine Unterscheidung zwischen motorisiert und nicht-motorisiert bei der ausschließlichen Verwendung von einzelnen Sensoren sehr gut funktioniert. Insbesondere lieferte der Beschleunigungssensor die besten Ergebnisse, um zwischen einem motorisierten und nicht-motorisiertem Zustand zu unterscheiden. Die Erkennungswahrscheinlichkeiten liegen dabei in den meisten Fällen über 90%. Die **Tabelle 1** fasst die wichtigsten Arbeiten in diesem Bereich zusammen.

Für die vorliegende Forschungsarbeit ist dabei von Interesse, mit Hilfe welcher Sensoren und mit welcher Genauigkeit motorisierte Zustände erkannt werden konnte. In der tabellarischen Aufstellung fällt auf, dass bei der Verwendung des gleichen Sensors wie beispielsweise GPS verschiedene Arbeiten zu unterschiedlichen

Erkennungswahrscheinlichkeiten von den gleichen Verkehrsmitteln kommen. Das liegt zum einen daran, dass für die Erkennung der Transportmittel unterschiedliche Verfahren des maschinellen Lernens verwendet wurden. Weiterhin sind die Qualität der Daten sowie das Testbett, also die Anzahl der Teilnehmer und die Dauer der Durchführung, entscheidend für die Transportmittelerkennung.

Quelle	Verfahren des maschinellen Lernens	Arten der Fortbewegung	Zustandserkennung [%]
(Gonzalez, et al., 2009)	Neuronale Netze (NN)	Gehen	100
		Bus	81,6
		Auto	92,1
(Xu, et al., 2010)	Fuzzy Logik (FL)	Gehen	96,4
		Fahrrad	88,9
		Bus Zug	93,2 57,1
(Xiao, et al., 2015)	Neuronale Netze (NN) Partikelschwarmoptimierung (PSO)	Gehen	98,5
		Fahrrad	87,5
		Bus	88,4
		Auto	94,3
(Xiao, et al., 2015)	Support Vector Classification (SVC) Genetischer Algorithmus (GA)	Gehen	100
		Fahrrad	100
		S-Bahn	88,9
		Bus Auto	92,7 80
(Lari, et al., 2015)	Random Forest (RF)	Bus	86,1
		Auto	97,4
		Gehen	90,8
(Zhang, et al., 2011)	Support Vector Machine (SVM)	Auto	100
		Bus	91
		Tram	78
		Zug	100
(Reddy, et al., 2010)	Decision Tree (DT) Hidden Markov Model (HMM)	Stehen	95,6
		Gehen	96,8
		Laufen	91,0
		Fahrrad	92,8
		Auto	93,9
(Manzoni, et al., 2011)	Decision Tree (DT)	Gehen	95
		Auto	91,25
		Fahrrad	88,75
		Bus	82,50
		Metro	73,75
		Zug	77,50

Tabelle 1: Gegenüberstellung der Erkennungswahrscheinlichkeiten für verschiedene Transportmodi

Für eine Transportmittelerkennung, wie diese im Rahmen eines intNAVs benötigt wird, ist jedoch eine ausschließliche Verwendung von einzelnen Sensoren nicht zu empfehlen, da bei einem intNAV die Unterscheidung innerhalb motorisierter Zustände von Interesse ist. In (Reddy, et al., 2010) wurde herausgefunden, dass bei der Verwendung eines einzelnen Sensors gegenüber einer Kombination von mehreren für die Transportmittelerkennung ein Einbruch um 10-20% bei der Erkennungswahrscheinlichkeit wahrnehmbar

ist. Aus diesem Grund widmen sich viele Arbeiten der Sensorfusion um bessere Transportmittelerkennungswahrscheinlichkeiten zu erzielen. Dies wird im nachfolgenden Kapitel besprochen.

2.2.2 Transportmittelerkennung mittels Sensorfusion

In einer Veröffentlichung von 2013 zeigte eine Forschergruppe um Feng eine verbesserte Transportmittelerkennung, wenn nicht nur ausschließlich der **GPS**-Sensor oder der **Accelerometer** verwendet werden, sondern wenn beide kombiniert werden (Feng, et al., 2013). Dabei fällt der Forschergruppe auf, dass der Accelerometer alleine den GPS-Sensor um Längen bei der Transportmittelerkennung übertrifft, jedoch eine Kombination aus beiden ein noch viel besseres Ergebnis liefert. Dabei wurde die Erkennungswahrscheinlichkeit der Fortbewegungsarten *gehen, laufen, Fahrrad, Motorrad, Bus, Auto, Zug, Metro* und *Tram* untersucht. Die Ergebnisse sind sehr vielversprechend. *Laufen, Fahrrad* und *Motorrad* erreichen eine Erkennungswahrscheinlichkeit von 100%, alle anderen erreichen eine Erkennungswahrscheinlichkeit von knapp 90%. Eine weitere Publikation, welche sich der Transportmittelerkennung mit Hilfe des **GPS**-Sensors und dem **Accelerometer** widmet, untersucht die Fortbewegungsarten *stehen, gehen, laufen, Fahrrad* oder *Auto* (Reddy, et al., 2010). Dabei wird mit Hilfe von DT und HMM die Erkennungswahrscheinlichkeit bestimmt. Dabei wurde eine Erkennungswahrscheinlichkeit von 93,6% erreicht. Der Kombination des **GPS**-Sensors sowie des **Accelerometers** widmet sich eine weitere Forschungseinheit um Xia aus China im Jahr 2014 (Xia, et al., 2014). Bei der Verwendung von SVM und Ant Colony Optimization (ACO) erreichen sie eine Erkennungsrate von 96,31% bei der Differenzierung der Transportmittel *gehen, Fahrrad* und *Auto*. 18 Benutzer testeten diese Lösung. Weiterhin untersucht eine Gruppe um Widhalm in Österreich, wie bei Verlusten von **GPS**-Signalen trotzdem gute Erkennungswerte mit der Kombination des **Accelerometers** erreicht werden können (Widhalm, et al., 2012). Mittels eines HMM sollen die Modi *gehen, Fahrrad, Motorrad, Auto, Bus, Tram, S-Bahn* und *U-Bahn* erkannt werden. Dabei haben 15 Probanden den entwickelten Ansatz zwei Monate lang getestet. Dabei konnten Erkennungswahrscheinlichkeiten von 91% erreicht werden. Anzumerken ist hier jedoch, dass das Testbett zu klein ist, um eine wirklich gute Aussage über die Erkennungswahrscheinlichkeit zu liefern.

Su et al. untersuchen die Güte der Transportmittelerkennung anhand der Sensoren **GPS**, **Accelerometer**, **Gyroskop** und **Magnetometer**. Dabei sollen folgende sechs Modi differenziert werden können: *Bus*, *U-Bahn*, *Auto*, *Fahrrad*, *gehen* und *laufen*. Zum einen lässt sich feststellen, dass eine 100% Differenzierung zwischen motorisierten und nicht-motorisierten Zuständen erreicht werden konnte. Zum anderen zeigen sie anhand von verschiedenen Ansätzen des Maschinellen Lernens (Bayes' Netz, Naive Bayes, RF, DT), dass die Transportmittelerkennung *Zug* mit fast 84%-iger Wahrscheinlichkeit, *Bus* 94%-iger, *Auto* 78%-iger, *Fahrrad* 100%-iger, *gehen* 99%-iger und *laufen* mit 78%iger Wahrscheinlichkeit erreicht werden kann. Dabei schneidet die Verwendung des Bayes' Netz am besten ab. Es sei zu erwähnen, dass die Daten im Nachhinein, also offline, bezogen und verarbeitet wurden, was bei einem intNAV nicht praktikabel wäre.

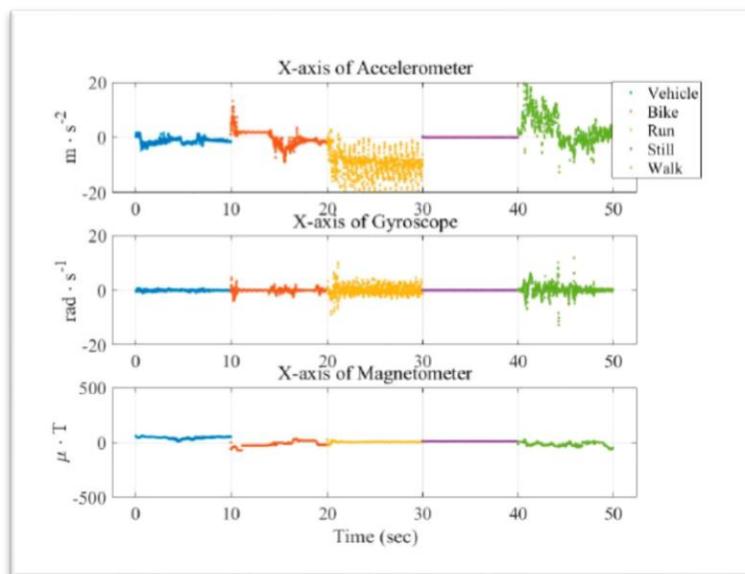


Abbildung 13: Verteilung der Roh-Daten auf der X-Achse von drei Sensoren zur Bestimmung des Transportmittels (Fang, et al., 2016)

Eine Forschungsgruppe aus Taiwan untersuchte eine Kombination aus drei Sensoren: **Accelerometer**, **Magnetometer** und **Gyroskop**, um eine Unterscheidung zwischen den nicht-motorisierten Zuständen *stehen*, *gehen*, *laufen*, *Fahrrad* und dem motorisierten Zuständen *Auto*, *Bus*, *Metro*, *Zug* und *Hochgeschwindigkeitszug* zu erhalten (Fang, et al., 2016). Dabei wurde ein Teil der von der Firma HTC zur Verfügung gestellten Daten von 224 freiwilligen Teilnehmern herangezogen, insgesamt 20 GB Daten. In **Abbildung 13** sieht man die unterschiedlichen Ausschläge bei der Verwendung eines Accelerometers, eines Gyroskops und eines Magnetometers für die Transportmittelerkennung. Mit Hilfe von drei Verfahren aus dem Bereich des Maschinellen Lernens (DT, KNN, SVM) konnte eine Erkennungswahrscheinlichkeit von durchschnittlich 83,57% bei dem KNN-Ansatz erreicht werden (60,62% bei DT und 78,59% bei SVM) für die Unterscheidung innerhalb der motorisierten Zustände *Metro*, *Bus*, *Auto*, *Zug* und *Hochgeschwindigkeitszug*. Um eine Unterscheidung von motorisierten und nicht-motorisierten Zuständen zu erhalten, wurden ebenfalls die drei erwähnten Verfahren aus dem Bereich des maschinellen Lernens herangezogen. Dabei wurde eine Erkennungswahrscheinlichkeit von

durchschnittlich 86,86% mittels dem KNN-Ansatz erreicht (79,59% bei DT und 86,94% bei SVM). Eine weitere Veröffentlichung, welche die Sensoren **Accelerometer**, **Mangetometer** und das **Gyroskop** für die Transportmittelerkennung heranzieht, untersucht die Transportmodi *gehen, laufen, Fahrrad und Auto* (Sonderen, 2016). Mit Hilfe der Algorithmen DT, Random Forest (RF) und KNN kann dabei mittels DT eine Erkennungswahrscheinlichkeit von 98% erreicht werden. Jedoch wird hier schnell deutlich, dass dabei nur ein motorisierter Zustand von zwei nicht-motorisierten Zuständen unterschieden werden soll. Dies ist im Sinne des intNAVs jedoch unbrauchbar.

2.2.3 Transportmittelerkennung mittels zusätzlicher Kontextinformation

Obwohl die aufgeführten Erkennungsraten aus dem vorangegangenen Kapitel bereits gute Resultate liefern, existiert in der Literatur ein dritter Forschungsansatz, um die Transportmittel möglichst genau erkennen zu können. Dabei soll, neben einzelnen Sensoren und Sensorfusion, zusätzlich Kontextinformation hinzugezogen werden um Transportmittel möglichst präzise zu erkennen. Wie sich zeigen lässt, bietet dieser Ansatz die besten Erkennungswahrscheinlichkeiten. Die wohl bekannteste Arbeit in diesem Zusammenhang ist von Stenneth et al. aus dem Jahr 2011 (Stenneth, et al., 2011). Die Forschergruppe untersucht neben dem **GPS**-Signal auch das in der Stadt zugrunde liegende **Verkehrsnetz**. Dabei wird eine Erkennungsrate von 93,5% für die Transportmittel *Auto, Bus, S-Bahn, Tram, gehen und Fahrrad* erreicht. Das Besondere an diesem Ansatz ist, dass dieser innerhalb der verfügbaren Literatur als einziger eine so hohe Erkennungsrate innerhalb der einzelnen motorisierten Zustände erreicht. Zusätzlich kann mit Hilfe des vorhandenen Algorithmus nicht nur sehr gut zwischen den einzelnen motorisierten Zuständen unterschieden werden. Im Gegensatz zu den bereits aufgeführten Arbeiten kann bei der vorliegenden Arbeit von Stenneth et al. zusätzlich die Bus-Linie, mit welcher der Benutzer unterwegs ist, ermittelt werden. Diese Arbeit von der Forschungsgruppe um Stenneth erfüllt dabei schon sehr gut die Anforderungen an ein intNAV. Dabei werden Ansätze des Maschinellen Lernens, wie Bayes' Netz, DT, RF, Naive Bayes und Multilayer Perception angewendet. Die erzielten Ergebnisse dabei sind beeindruckend. Zum ersten Mal wurde in der Literatur eine so gute Erkennungswahrscheinlichkeit **innerhalb** motorisierter Zustände erreicht. **Abbildung 14** zeigt die Ergebnisse der Publikation. Diese Arbeit war auch gleichzeitig die Inspiration für die vorliegende Dissertation. Es wurde nach Möglichkeiten gesucht, wie verschiedene Transportmittel in Echtzeit mittels Smartphone-Sensoren erkannt werden können und wie dabei zusätzlich weitere Kontextinformationen wie die Linie, aktuelle Haltestelle und die Richtung eines ÖPNVs zuverlässig erkannt werden können. Die Idee, zusätzliche Kontextinformation, in diesem Falle das zugrunde liegende Verkehrsnetz, heranzuziehen, um sehr gute Resultate zu erhalten, wurde bei der vorliegenden Dissertation aufgegriffen, jedoch mit dem Unterschied, dass ein regelbasiertes System verwendet wurde, welches bessere Erkennungsraten erzielte.

Mode	Duration (min)	Accuracy %
train	35	93
bus	30	95
car	30	89
walk	30	92
bike	30	93
stationary	34	98.5

Abbildung 14: Erkennungsraten von unterschiedlichen Transportmitteln nach Stenneth et al. (Stenneth, et al., 2011)

Interessant ist auch die Arbeit von Montaya et al. aus dem Jahr 2015 (Montoya, et al., 2015). Die Veröffentlichung zeigt die Verwendung von Kontextinformation zusätzlich zu den Sensordaten eines Smartphones, jedoch werden im Gegensatz zu der Arbeit von Stenneth et al. nicht nur ein Smartphone-Sensor (GPS) sondern mehrere (**GPS, Wi-Fi, Accelerometer** und **Bluetooth**) verwendet. Als Kontextinformation werden ähnlich wie bei der Arbeit von Stenneth et al. das zu Grunde liegende öffentliche Verkehrsnetz in Paris herangezogen, zusätzlich jedoch auch noch die Abfahrtszeiten des ÖPNVs. Das zu Grunde liegende öffentliche Verkehrsnetz wird dabei von Open Street Map (**OSM**) bezogen. Jedoch erfüllt die Arbeit von Montaya et al. die Anforderungen an ein intNAV nicht ganz. Es werden nur die Modi zu *Fuß, Fahrrad, Auto, Bus, Tram, Metro* und *Zug* differenziert. Um die Erkennungswahrscheinlichkeiten zu bekommen, werden Methoden des Maschinellen Lernens, dynamisches Bayes'sches Netzwerk, herangezogen. **Abbildung 15** zeigt die Erkennungsraten der einzelnen Transportmittel. Dabei fällt auf, dass wie auch bei Stenneth et al. Erkennungswahrscheinlichkeiten von über 75% innerhalb der einzelnen motorisierten Zustände (Auto, Bus, Zug und Tram) erreicht werden können. Interessanterweise wird jedoch deutlich, dass eine Sensorfusion aus unterschiedlichen Transportmitteln nicht zu einer Verbesserung der Transportmittelerkennung geführt hat. Ausschlaggebend für die gute Transportmittelerkennung scheint auch hier die Hinzunahme von zusätzlicher Kontextinformation gewesen zu sein.

		Predicted mode						Time
		foot	bike	car	bus	train	tram	
Actual mode	foot	87%	8%	1%	1%	2%	1%	1068'
	bike	2%	98%	0%	0%	0%	0%	69'
	car	5%	2%	82%	10%	0%	0%	718'
	bus	4%	5%	0%	90%	1%	0%	419'
	train	12%	0%	2%	3%	83%	0%	149'
	tram	15%	3%	6%	1%	0%	75%	129'
	Precision	91%	36%	96%	80%	81%	92%	2552'

Abbildung 15: Erkennungsrate der Transportmittel nach Montaya (Montoya, et al., 2015)

Auch J. Chen und M. Bierlaire aus Frankreich verwenden für die Transportmittelerkennung eine Sensorfusion aus **GPS, Wi-Fi, Bluetooth** und **Accelerometer** mit zusätzlichem Map-Matching auf dem zugrunde liegenden Verkehrsnetz (Chen, et al., 2015). Dabei wird die Wahrscheinlichkeit basierend auf den erhobenen Sensordaten für eine entsprechende Route auf dem Verkehrsnetz berechnet. Anschließend wird die Wahrscheinlichkeit für das entsprechende Transportmittel extrahiert. Im Gegensatz zu den vorgestellten Arbeiten erkennt der Algorithmus von Chen und Bierlaire automatisch die einzelnen Segmente einer multimodalen Route. Das Besondere an dem etablierten Algorithmus von Chen und Bierlaire ist, dass es die einzelnen unterschiedlichen Arten von Transportmitteln auf einer intermodalen Route erkennen kann. Die Daten, um automatisiert die multimodale Route in ihre einzelnen Segmente aufzuteilen, stammen dabei von den nahegelegenen Wi-Fi Zugangspunkten aber auch von sich in der Nähe befindlichen Bluetooth-Geräte. Leider wird in dieser Arbeit nicht erwähnt, welche Transportmittel erkannt wurden und mit welcher Erkennungsrate.

2.3. ZUSAMMENFASSUNG UND WEITERE SCHRITTE

Für ein intNAV ist die automatisierte Transportmittelerkennung in Echtzeit unabdingbar. Nur wenn das intNAV in der Lage ist, automatisiert die einzelnen Segmente einer Route zu erkennen (also in welchem (öffentlichen) Transportmittel sich der Benutzer innerhalb einer intermodalen Route befindet), kann eine Echtzeitnavigation erfolgen. Zusätzliche zu der TME in Echtzeit spielt aber auch die richtige Erkennung der Linie, Richtung und der einzelnen Haltestellen eines ÖPNVs eine wichtige Rolle bei einem intNAV. Die analysierten Arbeiten zur Transportmittelerkennung in Echtzeit lassen sich dabei unterteilen in Arbeiten, welche für die Transportmittelerkennung ausschließlich **einzelne Sensoren** verwenden, in Arbeiten, welche **mehrere Sensoren** verwenden, und schließlich in Arbeiten, welche zusätzlich **Kontextinformation** verwenden.

Zunächst ist auffällig, dass die meisten Arbeiten für die Transportmittelerkennung auf den GPS-Sensor zurückgreifen, sei es exklusiv oder in Kombination mit weiteren Sensoren und Kontextinformationen. Dabei werden Konzepte des Maschinellen Lernens herangezogen um die Art der Fortbewegung zu erkennen. Wenige Arbeiten konzentrieren sich auf Techniken des regelbasierten Ansatzes. Weiterhin ist auffällig, dass nur sehr wenige Arbeiten sich mit der Unterscheidung innerhalb motorisierter Zustände beschäftigen. Die meisten Arbeiten differenzieren die Transportmittel *gehen, Fahrrad, Auto* und *Bus*. Dies mag an den unterschiedlichen Anwendungsgebieten liegen, für ein intNAV ist jedoch die Unterscheidung zwischen einzelnen motorisierten Zuständen unabdingbar. Noch weniger Arbeiten jedoch widmen sich der automatisierten Extraktion der einzelnen Segmente auf einer multimodalen Route sowie der Erkennung der Transportmittel in Echtzeit. Auch dies ist eine kritische Anforderung für ein intNAV.

Am nächsten zu unserem Ansatz für die TME innerhalb eines intNAVs liegen die Arbeiten, welche zusätzlich zu den Sensordaten auch Kontextinformation verwenden, da diese eine automatisierte Segmentierung der intermodalen Route erlauben und die Erkennung der einzelnen Transportmittel am höchsten ist. Diese Ansätze liefern die Basis für das hier vorgestellte Konzept, welches in Kapitel 3 weiter vertieft wird.

3 INTERMODALE TRANSPORTMITTELERKENNUNG (TME) IM INTNAV

In diesem Kapitel wird ein Konzept präsentiert, mit welchem die Transportmittelerkennung (TME) innerhalb intermodaler Routen automatisiert in Echtzeit auf einem Android Smartphone erfolgen kann. Dabei wurde die zu Grunde liegende Infrastruktur mit Hilfe von Key-Points (1) und die TME in Echtzeit mittels einer oberen und unteren Logik umgesetzt (2). Die anschließende Testphase des entwickelten Algorithmus (3) zeigte eine Erfolgsquote von 96,77% der erkannten Transportmittel (sowohl motorisiert als auch nicht motorisiert) auf und ist damit einzigartig bei der automatisierten TME innerhalb von intNAVs.

[Dieses Kapitel basiert auf der (1) **Veröffentlichung** „*Dynamic Detection of Transportation Modes using Key-Point Prediction*“ (Olga Birth, Aaron Frueh, Johann Schlichter), Januar 2016 im Machine Learning, Optimization, and Big Data Volume 9432 of the series Lecture Notes in Computer Science pp. 49-59; (2) **Veröffentlichung** „*Socializing Public Transportation Using Situational Context in Public Transportation to Get in Touch with People Around You*“ (Michael Karg, Roman Roor, Olga Birth, Markus Strassberger), Mai 2015 auf der Konferenz *International Conference on Vehicle Technology and Intelligent Transport Systems*, at Lisbon, Portugal; (3) **Masterarbeit** „*Dynamic Detection of Transportation Modes using Key-Point Prediction*“ von Aaron Frueh, Betreuung: Olga Birth, Aufgabensteller: Prof. Dr. Johann Schlichter, abgenommen an der Fakultät für Informatik der TUM am 15. Dezember 2014]

Wie in **Abbildung 6** (Seite 18) zu sehen ist, bildet das Thema Transportmittelerkennung (TME) eines der zwei zentralen Themen der vorliegenden Arbeit. Die Notwendigkeit einer TME in Echtzeit wurde im vorhergehenden Kapitel ausführlich beleuchtet sowie die vorliegenden Arbeiten dazu. Dabei wurde außerdem die Tauglichkeit der verschiedenen Sensoren von Smartphones untersucht. Es erscheint sinnvoll für die vorliegende Arbeit das Smartphone zu verwenden, um darauf das intNAV zu erproben. Die Sensoren der aktuellen Smartphones sind in der Lage drei Modi mit sehr hoher Wahrscheinlichkeit zu bestimmen. Dazu zählen *Gehen*, *Öffentlich* und *Auto*. Nun kann argumentiert werden, dass es ausreicht zu wissen, ob ein Benutzer nun mit Hilfe eines dieser Modi unterwegs ist. Bei einem intNAV kann diese Genauigkeit aber damit verglichen werden, zu wissen, ob bei einem AutoNAV eine Person aktuell Richtung Norden oder Süden fährt. Für intNAVs ist die Differenzierung danach, ob es sich bei dem Status *Öffentlich* um einen Zug, eine S-Bahn, eine U-Bahn, eine Tram oder einen Bus handelt, *essentiell*. Aus diesem Grund, reicht die aktuelle sehr gute Differenzierung zwischen den Transportmodi *Gehen*, *Auto* und *ÖPNV* nicht aus.

3.1. TME MITTELS KEY-POINTS (KP)

In Hinblick auf das Kapitel 2.2, in welchem die Vor- und Nachteile verschiedener Smartphone-Sensoren sowie die Sensorfusion diskutiert wurden und die Anwendung des Maschinellen Lernens (**ML**) sowie Regelbasierter Systeme (**RBS**) auf diese Sensoren untersucht wurde, wurde für die vorliegende Arbeit ein Algorithmus entwickelt, welcher die TME mittels eines RBS sowie einer Sensorfusion mittels Kontextinformation verfolgt. Da eine der Anforderungen an das intNAV dessen Verwendung auf einem Smartphone ist und somit auch die TME auf einem Smartphone ausführbar sein muss, bedient sich der entwickelte Algorithmus einer Sensorfusion und der Zuhilfenahme von Kontextinformationen, da dort die Wahrscheinlichkeit für eine automatisierte TME in Echtzeit laut der Literaturrecherche am höchsten ist. Ein Abgleich der relevanten Arbeiten bezüglich der TME mittels Methoden des ML und RBS verdeutlichte, dass die Erkennungsrate bei RBS höher oder mindestens genauso hoch wie die bei Methoden des ML ist. Zusätzlich sind RBS flexibler als Ansätze des ML, da neue Regeln einfach hinzugefügt und entfernt werden können, ohne dass der gesamte Datensatz wie bei ML-Verfahren dafür neu trainiert werden muss. Diese Eigenschaft erweist sich als vorteilhaft, da wie eingangs beschrieben in verschiedenen Städten unterschiedliche Transportmittel im ÖPNV vorhanden sind und diese bei RBS flexibel hinzugenommen werden können.

Für das RBS wurde die Idee der Key-Points (**KPs**) umgesetzt. KPs sind mit Points-of-Interest (**POI**) zu vergleichen. Aufgrund der Literaturrecherche über die bestehenden Arbeiten wurde dieser Ansatz schnell favorisiert, da das Hinzufügen von Kontextinformationen die besten Resultate liefert. Der hier entwickelte Ansatz geht jedoch noch einen Schritt weiter als die in der Literatur beschriebenen Ansätze, welche mittels RBS über POI und der Verwendung von hybriden Sensoren sehr gute Erkennungsquoten erzielen. Der hier vorliegende Ansatz ist in der Lage nicht nur mit sehr guter Erkennungswahrscheinlichkeit das korrekte Fortbewegungsmittel zu erkennen, sondern auch noch weitere Information zu extrahieren wie z.B. die Erkennung der Linie, der Richtung, der einzelnen Stationen und auch der Ein- und Ausgänge zu den Stationen. Während die in der Literatur genannten Algorithmen sehr gute Treffer liefern, müssen diese bei der TME jedoch für die Verwendung im intNAV angepasst werden. Für das intNAV muss die TME daher in der Lage sein, folgende Punkte in Echtzeit zu erfüllen:

Transportmittelerkennung (TME):

Automatisierte Differenzierung einzelner motorisierter und nicht-motorisierter Modi auf den einzelnen Routenabschnitten r_i 's der Route R.

Erkennung von Detailinformationen des jeweiligen Transportmittels (Typ, Linie/Richtung, Haltestellen)

Ausführung auf dem Smartphone

Wie eingangs erwähnt ist die TME einer von zwei Grundpfeilern der intermodalen Navigation. Der zweite Grundpfeiler ist die Erkennung der Deviation. Für diese jedoch wird eine Transportmittelerkennung (**TME**) auf der beschriebenen Granularität benötigt. In diesem Kapitel wird der entwickelte Algorithmus für die Transportmittelerkennung beschrieben, welcher diesen Anspruch genügt.

3.1.1 Konzept von KPs

KPs stellen eine einfache Repräsentation der Welt dar. KPs sind Zusatzinformationen mit deren Hilfe mittels RBS Rückschlüsse gezogen werden können, welches Fortbewegungsmittel eine Person gerade verwendet. Dabei wird das Konzept der logischen Schlussfolgerungen eines wissensbasierten Systems verwendet. So können beispielsweise mit Hilfe der KPs Schlussfolgerungen über die zugrunde liegende Infrastruktur eines öffentlichen Netzes einer Stadt geliefert werden. Auch können KPs Aussagen darüber liefern, ob eine Person gerade mit einem Auto fährt oder zu Fuß geht. Am wichtigsten ist jedoch die folgende Eigenschaft eines KPs. Mit Hilfe eines KPs kann mit sehr hoher Wahrscheinlichkeit nicht nur das korrekte Transportmittel identifiziert werden, es können sogar weiterführende Detailinformationen wie die Linie, die Richtung, Zugänge und die nächsten Stationen in Echtzeit extrahiert werden.

KPs sind *Orte*, welche Rückschlüsse über zusätzliche Kontextinformation oder logische Information zulassen. Folgende Anforderungen an (eine Menge von) KPs müssen dabei erfüllt sein:

Flexibel

Die zu Grunde liegende Menge an KPs ist beliebig erweiter- bzw. verfeinerbar. Der hier vorgestellte Ansatz soll überall angewendet werden können, nicht nur in der Stadt, in welcher diese Lösung erarbeitet und deren Funktionalität getestet wurde.

Symbolisch/Numerisch

Die Datenrepräsentation der KPs soll sowohl symbolische Werte als auch numerische Werte annehmen können.

Transparent

Die Repräsentation der zugrunde liegenden Weltinformation soll einfach und explizit sein, sodass sie nachvollziehbar ist.

Korrelationen

Es müssen Korrelationen zwischen den KPs vorhanden sein. Das System muss in der Lage sein, logische Entscheidungen basierend auf Abhängigkeiten zwischen Daten zu fällen.

Es existieren keine weiteren Einschränkungen für einen KP. Folgende **Tabelle 2** bietet einen Auszug über mögliche Arten von KPs. Nicht alle wurden in der vorliegenden Arbeit verwendet.

Typ	Logische Rückschlüsse über den KP
Stationen/ Haltestellen	Die Orte von Stationen und Haltestellen wie z.B. Bushaltestellen, Tramhaltestellen oder Taxistationen können sinnvolle logische Schlüsse bezüglich der Infrastruktur des öffentlichen Verkehrsnetzes bieten.
W-LAN Zugangspunkte	Zugangspunkte können in Verbindung mit anderen KPs gebracht werden. Dies entspricht dem Konstrukt von Sensorfusion und kann zu einer besseren Erkennungswahrscheinlichkeit führen.

Park & Ride	Diese Ortsinformation kann Rückschlüsse über einen Transportwechsel darstellen.
Eingänge/Fahrstühle	Unterirdische Positionierungen von ÖPNVs sind meistens irreführend und sinnlos. Die Positionen von Eingängen/Fahrstühlen zu Stationen liefern einen besseren Hinweis darauf, ob und welches öffentliche Verkehrsmittel verwendet wird.
Knotenpunkte	Ein Knotenpunkt kann dabei behilflich sein, zwischen Autos und Bussen besser unterscheiden zu können. So können Fahrzeuge, welche an einer bestimmten Stelle (Knotenpunkt) wenden, auf einen Bus hinweisen, wenn dies mit der entsprechenden Route quergecheckt wurde.
Variable Punkte	Variable Punkte können z.B. der Abstellort des Fahrzeuges sein. Die Erkennung könnte mittels GPS oder weiteren Sensoren wie sich in der Nähe befindlichen WLAN Zugangspunkten erfolgen.

Tabelle 2: Überblick über mögliche KP-Typen und die Rückschlüsse bezüglich der TME

Solange ein KP Rückschlüsse über ein bestimmtes Transportmittel zulässt, ist es ein gültiger KP. In dieser Arbeit wurden KPs mit Hilfe von Haltestellen von öffentlichen Verkehrsmitteln sowie der Ein- und Ausgänge und Knotenpunkte modelliert und keine weiteren Kontextinformationen (vgl. **Tabelle 2**) herangezogen. Die Haltestelleninformation sowie die Zugänge wurden aus öffentlich verfügbaren Quellen für Kartendienste (Open Street Map) bezogen. Dabei repräsentiert der Bezug der KPs untereinander die öffentliche Infrastruktur. Diese Information wurde auf einen gerichteten Graphen übertragen, wobei die Knoten die einzelnen Haltestellen darstellen und jeder Knoten durch mindestens eine Kante verbunden werden kann. **Abbildung 16** zeigt einen solchen Graphen. Die entsprechende reale Repräsentation auf der Karte kann dem Anhang entnommen werden.

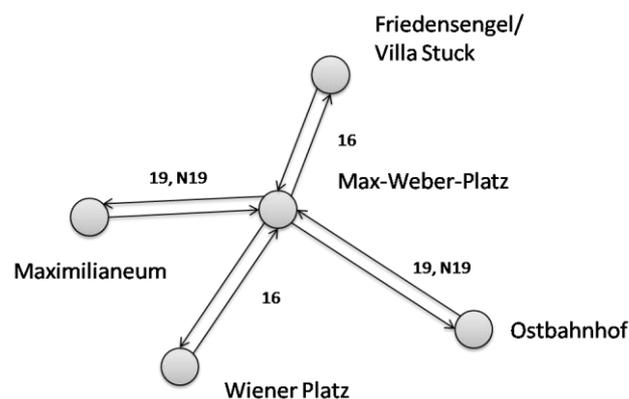


Abbildung 16: Abstrakte KP-Graphen-Visualisierung und Darstellung des Bezugs zwischen mehreren KPs, um die öffentliche Infrastruktur in München abzubilden

Die **Abbildung 16** zeigt die Tramverbindungen an der Station Max-Weber-Platz in München. Somit wird jede (Bus-, Tram-, S- und U –Bahn) Station durch einen KP repräsentiert. Wie in **Tabelle 2** beschrieben ist, werden auch die Ein- und Ausgänge zu den jeweiligen Stationen mittels KPs repräsentiert. Dies ist auf dem Graphen in

Abbildung 16 nicht visualisiert. Es gibt beispielsweise U-Bahn Stationen, die mehrere Ein- und Ausgänge besitzen. Jeder einzelne davon wird bei einer abstrakten Graphen-Repräsentation als ein KP repräsentiert. Die Fusion mehrerer KPs zu einem größeren wird auch KeyPointCluster (**KPC**) genannt. Diese Cluster werden verwendet, um den ursprünglichen Punkt zu identifizieren. Der Grund hierfür ist, dass erst die Zugänge zu einer Station Rückschlüsse darüber liefern, ob eine Person an der Station ist und es sich dabei nicht um die geographische Position einer Station handelt. Im Kapitel 3.1.2, in welchem der Entscheidungsprozess des RBS erläutert wird, wird der Grund für diese akribische Repräsentation jedes einzelnen Ein- und Ausgangs beschrieben.

Der hier entwickelte Algorithmus zeigt also hohe Wahrscheinlichkeiten für sich in der Nähe befindliche KPs und niedrige Wahrscheinlichkeiten für weiter entfernte KPs an. Der Grund hierfür ist, dass ein Benutzer, welcher sich einer Haltestelle nähert, diese auch ansteuern wird. Somit sind sämtliche KPs im Umkreis des Benutzers mit einer hohen Wahrscheinlichkeit versehen. Für die Berechnung wird nicht die echte, sondern die lineare Distanz verwendet. Der Grund hierfür ist, dass die Berechnung zwar mit der echten Distanz akkurater wäre, aber der entwickelte Prototyp offline benutzt werden soll. Zusätzlich würde sich der Berechnungsaufwand signifikant erhöhen, abhängig von der Verkehrsnetzichte.

Die folgenden Gleichungen beschreiben wie sich die Berechnung der Wahrscheinlichkeiten zusammensetzt. Die Variable d_i stellt dabei die Distanz zwischen dem aktuellen Ort und dem KP (für jeden KP) dar. n beschreibt dabei sämtliche KPs, wobei in Gleichung 3.4 eine Begrenzung nach oben eingeführt wurde.

$$sumDistance = \sum_{i=0}^n d_i \quad (3.1)$$

$$I(d) = \left(\frac{d}{sumDistance} \right)^{-1} = \frac{sumDistance}{d} \quad (3.2)$$

Die Gleichung 3.2 berechnet die Inverse der Distanzverteilung zwischen allen KPs. Die größte Wahrscheinlichkeit erhält dabei der nächstgelegene KP und die niedrigste Wahrscheinlichkeit der weiteste KP. Alleinstehend liefert dies keine befriedigende Wahrscheinlichkeit, denn, wenn es auf eine hohe KP-Dichte, wie diese in München vorhanden ist, angewendet wird, liefern alle KPs niedrige Wahrscheinlichkeiten. Aus diesem Grund wurden Gewichtungen hinzugenommen (vgl. Gleichung 3.3).

$$A(d) \in]0,2] \quad (3.3)$$

$$A(d) = \begin{cases} \frac{\Delta d}{d_{t-1}} + 1 = \frac{d_{t-1} - d_t}{d_{t-1}} + 1, & \text{if } \Delta d > -d_{t-1} \\ 1, & \text{otherwise} \end{cases}$$

Die Gleichung 3.3 liefert ein differentiales Gewicht. KPs, welchen sich der Benutzer schneller annähert, werden höher gewichtet, als KPs zu welchen die gleiche Distanz bestehen bleibt. Die Wahrscheinlichkeit für KPs, welche sich entfernen wird dadurch signifikant reduziert. In der Gleichung 3.3 wird die Entfernung der letzten Messung und der aktuellen Distanz zwischen KPs verwendet. Dies macht es anfällig für Oszillationen. Der Durchschnitt über die letzten drei oder fünf Distanzen würde bessere Ergebnisse liefern.

Durch Ausprobieren wurde herausgefunden, dass die Wahrscheinlichkeit für KPs ab 200 Meter Entfernung nicht mehr ins Gewicht fällt. Aus diesem Grund wurde Gleichung 3.4 eingeführt, welche KPs unter 200 m berücksichtigt und danach alle die gleiche niedrige Wahrscheinlichkeit erhalten. Alles was über dieser Entfernung liegt, wird mittels der Funktion 3.4 neutralisiert.

$$B(d) \in [1,10] \tag{3.4}$$

$$B(d) = 1 + 9 * e^{-d*0.026347}$$

Die Gleichung 3.5 liefert nun die finale Gewichtung eines KPs. Dividiert durch das Gewicht aller KPs erhält man nun die Wahrscheinlichkeit (vgl. Gleichung 3.7).

$$W(d) = (I(d) * A(d) * B(d)) \tag{3.5}$$

$$S = \sum_{i=0}^n W(d_i) \tag{3.6}$$

$$P(d) = 100 * \frac{W(d)}{S} \tag{3.7}$$

Mittels Gleichung 3.7 können relevante KPs in der Nähe des Benutzers angezeigt werden, da KPs mit höherer Wahrscheinlichkeit auch ein wahrscheinlicheres Ziel darstellen. Die vorgestellten Gleichungen haben den Vorteil, dass sie sehr simpel auszuwerten sind, aber dennoch effektiv für das Konzept der Key-Points geeignet sind. Aus diesem Grund sind die Berechnungen sehr gut geeignet, um auf mobilen Endgeräten ausführbar zu sein.

3.1.2 Entscheidungsprozess

Dieses Kapitel behandelt den entwickelten Algorithmus und, wie dieser logische Entscheidungen trifft, um den Anforderungen an eine TME gerecht zu werden. Da der Anspruch an die vorliegende Arbeit ist, sehr genaue Informationen über die aktuelle Aktivität des Benutzers und somit des damit zusammenhängenden Transportmittel zu erhalten, ist es sehr wichtig, die Herausforderungen an ein solches System zu kennen. Dazu wurden zunächst Problemklassen identifiziert, welche relevante Transitionen aufzeigen, also mögliche Szenarien des Transportmittelwechsels auf einer Route. Im Fokus der vorliegenden Arbeit liegen dabei sämtliche Transitionen, welche mit dem ÖPNV in Zusammenhang stehen. **Tabelle 3** gibt einen Überblick über die möglichen Transportmittelwechsel.

	nach			
von		Gehen	Auto	ÖPNV
Gehen	X	A	B	
Auto	A	X	C	
ÖPNV	B	C	D	

Tabelle 3: Mögliche Transitionen zwischen den Modi

Klasse X

Die Transitionen zwischen den Status Gehen und Gehen sowie zwischen Auto und Auto sind entweder trivial oder logisch nicht möglich.

Klasse A

In dieser Arbeit müssen Transitionen erkannt werden, welche den Übergang zwischen Gehen und Auto (und umgekehrt) darstellen. Die Herausforderung dieser Problemklasse ist dabei zwischen den verschiedenen motorisierten Zuständen zu unterscheiden (vergleiche dazu Klasse C).

Die nachfolgenden Problemklassen beschreiben den Fokus der vorliegenden Arbeit, da dabei der *Transportmittelwechsel* mit und innerhalb des ÖPNVs vorliegt. In der durchgeführten Literaturrecherche wurde hierbei die größte Forschungslücke aufgedeckt.

Klasse B

Problemklasse B und D ähneln sich zum Teil in der Logik. Dennoch ist die Problemklasse B so zu verstehen, dass eine „echte“ Transition zwischen dem Modus Gehen und dem Modus ÖPNV (und zurück) stattfindet und nicht der Modus Gehen ein Hilfsmodus ist, um zwischen einzelnen ÖPNV-Transportmitteln zu wechseln. Wie die konkrete Erkennung funktioniert, wird weiter unten beschrieben.

Klasse C

Die Klasse C ähnelt der Problemklasse B, wobei im Gegensatz zu Klasse B die Differenzierung zwischen Auto und ÖPNV hierbei im Vordergrund steht.

Klasse D

Um zwischen den einzelnen Fortbewegungsmitteln innerhalb des ÖPNVs zu wechseln, muss eine Person dabei ein Transportmittel verlassen und dann in das nächste einsteigen. Die Herausforderung innerhalb dieser Klasse ist es zum einen, zu erkennen, dass ein ÖPNV-Transportmittel *verlassen* wurde, und den *korrekten Zeitpunkt* zu identifizieren, wann dies geschehen ist. Manchmal findet ein solcher Transportmittelwechsel an einem Ort und innerhalb weniger Minuten statt. Aus diesem Grund wurde ein Hilfsstatus in der Logik eingeführt, der *Transitionsstatus*, um die Erkennung der Problemklasse D zu erleichtern. Da dies den schwierigsten Fall des entwickelten Algorithmus darstellt, verdient dieser Sachverhalt eine eigene Problemklasse.

Wir unterscheiden zwischen einer **oberen** und einer **unteren** Logik. Die untere Logik läuft innerhalb jedes einzelnen KPs ab, wohingegen die obere Logik die Information über alle KPs hinweg zusammenhält und basierend auf den Ergebnissen der unteren Logik Entscheidungen über die aktuelle Situation fällt.

3.1.2.1 OBERE LOGIK

Die obere Logik besteht aus insgesamt sechs Zuständen, wie in **Abbildung 17** dargestellt ist. Die primären Zustände sind dabei *Gehen*, *Auto* und *ÖPNV*. Zusätzlich gibt es noch sogenannte Hilfszustände wie *Unbekannt*, *Motorisiert* und *Transition*. Die Hilfszustände bilden keine echten Zustände ab, sondern werden benötigt, um mit deren Hilfe die richtigen Zustände, welche im Rahmen der TME benötigt werden, zu erkennen. Nachfolgend wird die Logik der einzelnen Transitionen zwischen den Zuständen im Detail diskutiert.

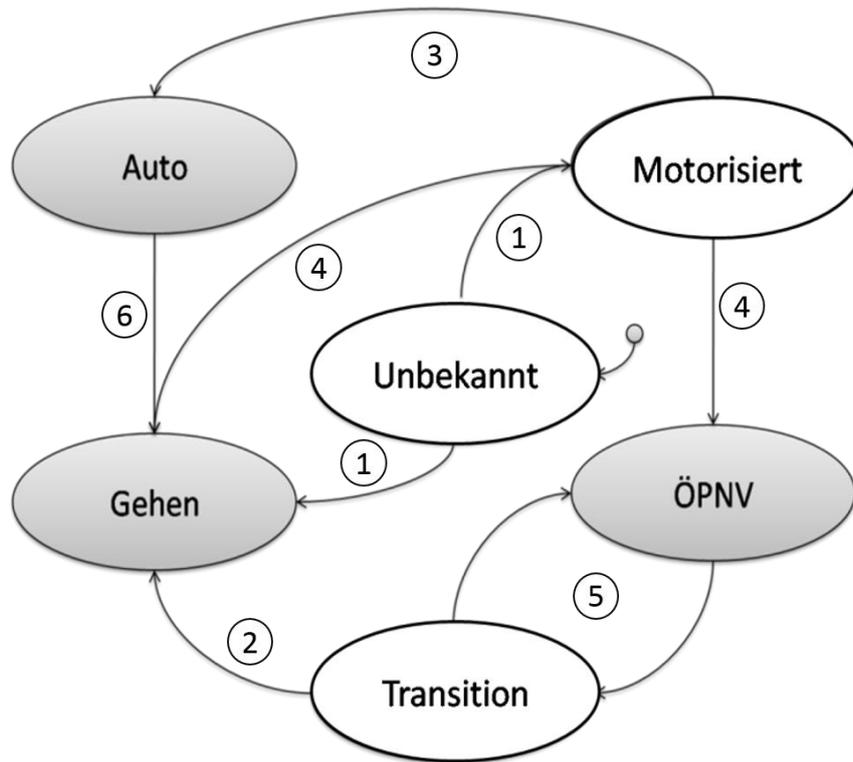


Abbildung 17: Zustandsautomat der oberen Logik

[Initialzustand] Unbekannt → Gehen/Motorisiert

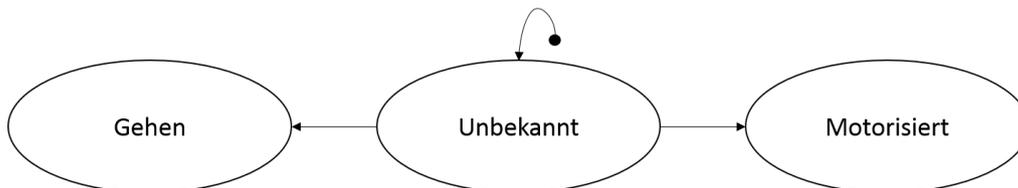


Abbildung 18: Logik: Unbekannt --> Gehen/ Motorisiert

Der initiale Zustand ist nicht weiter spezifiziert, was bedeutet, dass sich der Benutzer zum Startpunkt der TME in jedem vorstellbaren Zustand befinden könnte. Der erste Schritt der Logik ist es also herauszufinden, ob der Benutzer sich nun im Zustand Gehen oder im Zustand Motorisiert befindet. Wenn der Benutzer sich überhaupt nicht bewegt, wartet der Zustandsautomat so lange, bis eine Bewegung festgestellt wird. Da der entwickelte Prototyp für die TME auf einem Smartphone getestet werden soll, war die Überlegung naheliegend, für die Differenzierung des initialen Status die entsprechenden Smartphone-Sensoren zu verwenden, um die durchschnittliche Geschwindigkeit zu messen. Parallel dazu wurde die *Google Activity Recognition* getestet. Diese bereitgestellte Funktion implementiert dabei auf Basis des Beschleunigungssensors eine Differenzierung zwischen den Zuständen *Gehen, Laufen, zu Fuß, Fahrrad, Auto* und *Stehen*. Eine hohe Wahrscheinlichkeit für den Zustand *Auto* führt dazu, dass die obere Logik in den Hilfszustand *Motorisiert* übergeht. Obwohl in der *Google Activity Recognition* der Zustand *Auto* festgestellt wurde, fehlt weiterhin die für das intNAV benötigte

Differenzierung der einzelnen motorisierten Zustände wie Bus, Tram und S-Bahn. Dagegen werden die erkannten Zustände der *Activity Recognition* wie *Laufen*, *Gehen* oder *zu Fuß* zu dem echten Zustand der oberen Logik *Gehen* zusammengefasst. Für die Erkennung des Status *Gehen* und des Hilfsstatus *Motorisiert* wird auf Grund der sehr guten Erkennungsrate auf die *Google Activity Recognition* Funktion zurückgegriffen.

Aus dem Initialen Status **Unbekannt** wird der Zustand **Gehen** und der Hilfszustand **Motorisiert** mittels *Google Activity Recognition* erreicht.

Während sich der Zustandsautomat im Zustand *Unbekannt* oder *Gehen* befindet, wird die Umgebung nach KPs abgesucht und diese werden in die Liste möglicher Punkte aufgenommen. Diese Punkte werden entfernt, sobald der Benutzer den Suchradius verlassen hat, und gelten somit nicht als potentieller ÖPNV.

2. [Abbruchbedingung] Transition → Gehen

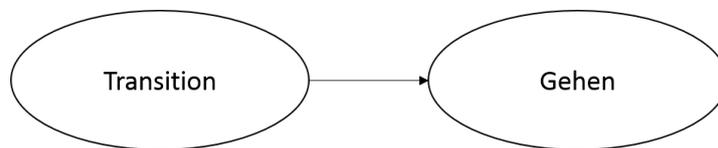


Abbildung 19: Abbruchbedingung

In jeder Transitionsphase werden die Abbruchbedingungen erneuert. In der erarbeiteten Logik sind nachfolgende drei Abbruchbedingungen implementiert:

Zeitüberschreitung

Sinkende Wahrscheinlichkeit des Ziel-KPs

Ergebnis der Google Activity Recognition

Bei der ersten Abbruchbedingung, der Zeitüberschreitung, nimmt der Algorithmus die Entfernung zu dem nächsten KP und schätzt einen maximalen Zeitrahmen, bis wann dieser KP erreicht werden kann. Diese Zeit wird gemessen, sobald der Benutzer den aktuellen KP-Bereich verlässt (**Abbildung 25**). Diese Abbruchbedingung ist zugleich eine Backup-Funktion, wenn alle anderen Abbruchbedingungen nicht ausgelöst wurden.

Die zweite Abbruchbedingung wird ausgelöst, wenn die Wahrscheinlichkeit für einen anvisierten KP sinkt. Die Berechnung der Wahrscheinlichkeit führt dazu, dass diese signifikant sinkt, sobald der Benutzer sich in eine andere Richtung bewegt. Sobald diese Wahrscheinlichkeit unter einen bestimmten Schwellwert fällt, welcher ebenfalls mit Hilfe der Distanz berechnet wird, kommt diese Abbruchbedingung zum Tragen. Diese Abbruchbedingung kommt leider nicht zum Tragen, wenn der Benutzer an einer Haltestelle umsteigt, weil die KPs zu nah beieinander liegen. Während der Entwicklungsphase wurde jedoch der optimale Bereich für den Radius und die Distanzberechnung festgelegt und zusätzlich andere Abbruchbedingungen definiert.

Die *Activity Recognition* von Google erlaubt eine dritte Abbruchbedingung. Da wie weiter oben beschrieben, die Erkennung der Status *Gehen* und *Auto* mit Hilfe dieser Funktion von Google sehr gut funktioniert, wird dies

ausgenutzt. Wenn also die *Activity Recognition* den Status Gehen oder Auto erkennt, dann wird diese Abbruchbedingung verwendet. Diese Abbruchfunktion wird nur im Status *Transition* abgefragt, da diese bedingt durch die Beschleunigung von den Transportmitteln im ÖPNV verfälschte Informationen liefern würde. Während der Umsetzungsphase wurde beispielsweise festgestellt, dass die Vibrationen in der Tram die Erkennung des Zustandes *Gehen* auslösen.

Mit Hilfe der *Google Activity Recognition* gelangt der Algorithmus aus dem Hilfszustand **Transition** in den Zustand **Gehen**.

3. [Zustandsübergang] Motorisiert → Auto

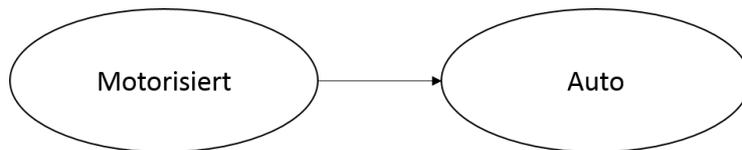


Abbildung 20: Logik: Motorisiert --> Auto

Die Differenzierung zwischen den einzelnen motorisierten Zuständen ist wie bereits erwähnt sehr schwierig. Um dieses Problem zu lösen, verwendet der entwickelte Algorithmus die KPs. Wenn sich der Algorithmus im Hilfszustand *Motorisiert* befindet und die *Google Activity Recognition* den Zustand "Auto" mit einer Wahrscheinlichkeit über dem Mittelwert (über 60%) erkennt, erfolgt der Zustandsübergang von *Motorisiert* auf *Auto*. Innerhalb dieses Zustandes wird die Umgebung nicht mehr nach neuen KPs überprüft, sondern es wird nur noch mit dem aktuell sich in der Liste möglicher KPs befindlichen Punkten weitergearbeitet (vgl. Abschnitt 3.1.2.2 Untere Logik). Sobald alle möglichen KPs ausgeschieden sind, was bedeutet, dass der jeweilige zuständige Zustandsautomat die Entscheidung getroffen hat, dass die Wahrscheinlichkeit für ein ÖPNV sehr gering ist, liefert die Obere Logik die Entscheidung zurück, dass der Benutzer sich gerade in einem Auto befindet.

Der Übergang vom Zustand *Motorisiert* zum Zustand *Auto* erfolgt, sobald die **Activity Recognition** von Google eine Wahrscheinlichkeit für den Zustand *Auto* von **über 60%** anzeigt und die **Zustandsautomaten** (aus der Unteren Logik) der jeweiligen **KPs kein gültiges ÖPNV** feststellen können.

4. [Zustandsübergang] Gehen → Motorisiert → ÖPNV

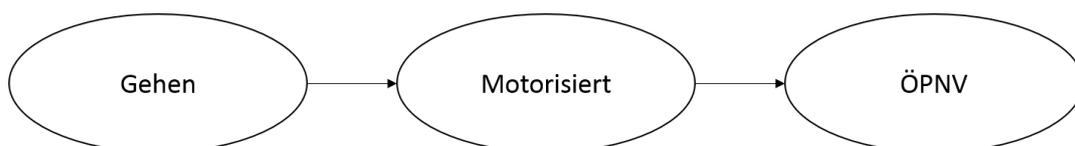


Abbildung 21: Logik: Gehen --> Motorisiert --> ÖPNV

Die Transition zu dem Zustand ÖPNV kann am besten mit der **Abbildung 22** beschrieben werden, worin am besten der Ablauf der Unteren Logik abgebildet ist.

Jeder KP besitzt einen eigenen Zustandsautomat, die untere Logik. Wenn ein Benutzer sich einem der hinterlegten KPs nähert, sendet die untere Logik eine Benachrichtigung an die obere Logik. Die **Abbildung 25** verdeutlicht dieses Konstrukt. Jeder KP, in **Abbildung 22** durch einen ausgefüllten Kreis dargestellt, ist entsprechend der ÖPNV-Infrastruktur mit anderen KPs durch gerichtete Pfeile verbunden. Zusätzlich besitzt jeder KP zwei Radien, einen kleineren (in **Abbildung 22** durch den Buchstaben **A** identifiziert) und einen größeren (in **Abbildung 22** durch den Buchstaben **B** identifiziert). Wenn der Benutzer sich nun im Kreis mit Radius **A** befindet und somit in der unmittelbaren Nähe des entsprechenden KP, sendet die untere Logik eine Benachrichtigung an die obere Logik. Zu diesem Zeitpunkt gibt es zwei Möglichkeiten: Entweder der Benutzer passiert den KP (Ein- und Ausgänge von den Stationen/ Haltestellen) oder der Benutzer steigt tatsächlich in das ÖPNV, welches durch den KP assoziiert wurde, ein. Sobald der Benutzer den Kreis mit Radius **B** verlässt, werden alle Kanten mit den dazugehörigen KP weiter verfolgt. Durch Ausprobieren während der Entwicklungsphase, wurde ein passender Radius für den Kreis **A** von 50 m und für den Kreis **B** von 75 m festgelegt, welcher die angesprochenen Anforderungen erfüllt.

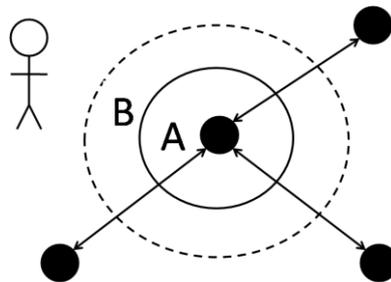


Abbildung 22: KP-Verhalten der unteren Logik

Mit der vorher festgelegten Annahme, dass jedes Transportmittel des ÖPNV an den entsprechenden Stationen hält, werden alle KP, welche in der ÖPNV-Infrastruktur zusammenhängen, benachrichtigt und in der Liste der in Frage kommenden KP hinterlegt. Die jeweiligen Zustandsautomaten der einzelnen KPs verfolgen nun, ob der Benutzer sie ansteuert oder nicht. Wenn die Wahrscheinlichkeit für einen ausgewählten KP fällt, wird dieser aus der Liste aussortiert und der KP kehrt in den initialen Zustand zurück.

Wenn nun der Benutzer den nächsten KP erreicht, wird mittels der Smartphone-Sensoren darauf geachtet, ob die Geschwindigkeit unter einen Schwellwert von fünf Kilometer pro Stunde fällt. Dieser Schwellwert wurde ebenfalls in Zuge der Entwicklung des Prototypen festgelegt. Zu diesem Zeitpunkt kann nun eine Entscheidung erfolgen und alle Linien, welche zu diesen Kanten gehören, werden in die Kandidatenliste der möglichen ÖPNV-Linien aufgenommen. Da mehrere Linien zu einer gerichteten Kante im Graphen gehören können, besitzen alle die gleiche Wahrscheinlichkeit. Die Linien behalten solange die gleiche Wahrscheinlichkeit, wie die Kante die gleiche ist. Sobald jedoch Linien abzweigen und somit nicht mehr zu der Kante korrespondieren, erhalten sie die niedrigste Wahrscheinlichkeit gegenüber den anderen Linien, welche zu der Kante korrespondieren. Wenn

Echtzeitabfahrtszeiten zur Verfügung stehen, kann der Abgleich damit die Linien eliminieren, welche nicht mit der nächsten Abfahrtszeit übereinstimmen.

Die **automatisierte** Erkennung der Transition zwischen den Zuständen *Gehen* und *ÖPNV* erfolgt mit Hilfe des **Zustandsautomaten** der **KP** (Untere Logik).

5. [Zustandsübergang] ÖPNV → Transition → ÖPNV

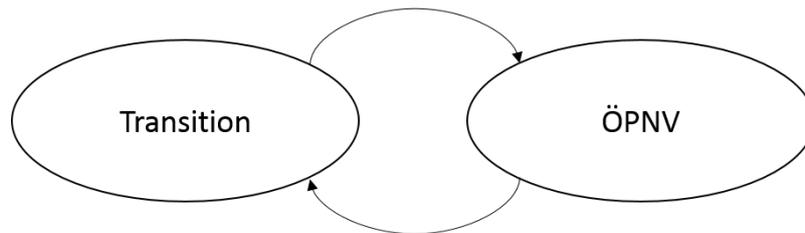


Abbildung 23: Logik ÖPNV <--> Transition

Der Zustand *Transition* ist wie der Zustand *Unbekannt* und *Motorisiert* ein Hilfszustand, welcher benötigt wird um den Wechsel innerhalb eines *ÖPNV* zu erkennen (vgl. hierzu auch die Problemklasse D aus **Tabelle 3**). Die untere Logik beschreibt dabei das Vorgehen, wie festgestellt werden kann, ob eine Transition innerhalb eines *ÖPNV*s stattgefunden hat oder nicht. Die obere Logik dient lediglich in diesem Status dazu, mögliche Kandidaten für die Transitionen aufzunehmen oder zu eliminieren.

Während der Ziel-KP anvisiert wird, verfolgt die Logik den gerichteten Graphen und setzt das Ziel auf den sich auf der Kante am nächsten befindlichen KP. Dabei priorisiert es die aktuellen möglichen Linien. Wenn jedoch neue Linien hinzukommen, weil neue KPs mit dazugehörigen Kanten in die Liste hinzugenommen wurden, werden diese weiteren Linien im Hintergrund weiterverfolgt, weil der Benutzer auf diese wechseln kann. In **Abbildung 24** wird graphisch die beschriebene Logik erklärt. Der aktuelle Punkt (AP) wird zum Zielpunkt (ZP). Die durchgehenden Linien kennzeichnen die möglichen Umsteigepunkte (Transitionen). Die gestrichelten Linien zeigen weitere Kanten an, welche von den Transitionen weggehen. Diese werden im Hintergrund weiter verfolgt solange, bis diese ausgeschlossen werden, weil z.B. eine Transition auf diese Kanten nicht mehr möglich ist.

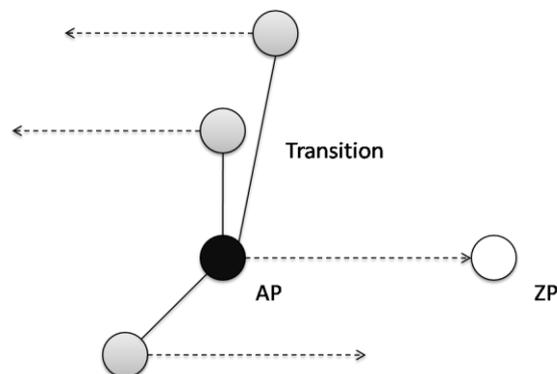


Abbildung 24: Mögliche Transitionen (AP = Aktueller Punkt; ZP = Zielpunkt)

Die obere Logik des Status *Transition* wacht über die mögliche **Kandidatenliste** der Umsteigepunkte.

Anhand des eingeführten Zustandsautomaten in **Abbildung 17** wurde die obere Logik beschrieben. Die obere Logik dient dazu, zum einen die präsentierten Problemklassen zu behandeln, und zum anderen, um die „echten“ Zustände wie *Gehen*, *ÖPNV* und *Auto* mit Hilfe der Hilfszustände *Transition*, *Unbekannt* und *Motorisiert* zu unterscheiden. Dabei wurde festgestellt, dass mit Hilfe der *Activity Recognition* von Google zuverlässige Differenzierung zwischen den Status *Gehen* und *Auto* vorgenommen werden können, dass jedoch eine Lösung entwickelt werden muss, welche den Zustand *ÖPNV* korrekt erkennen und auch die einzelnen verschiedenen Transportmittel mit den jeweiligen Kontextinformationen darin erkennen kann. Zu diesem Zweck wurde die untere Logik eingeführt. Die obere Logik greift dabei auf die Erkenntnisse der unteren Logik zurück und liefert eine Wahrscheinlichkeit für das gerade erkannte Transportmittel.

3.1.2.2 UNTERE LOGIK

Die nachfolgende **Abbildung 25** zeigt die Zustände der unteren Logik. Dabei handelt es sich um einen Zustandsautomaten, welchen jeder einzelne KP besitzt und welcher den erkannten Zustand an die obere Logik sendet. Bei dem vorangehenden Kapitel wurde mittels der KP-Logik der Statuswechsel von *Gehen* → *Motorisiert* → *ÖPNV* erklärt. Dabei wurde mit Hilfe zweier Radien (**A** und **B**) die Erkennung definiert, wann ein KP als „nächster KP“ markiert wird. Der KP, zu dem der nachfolgende Zustandsautomat dargestellt ist, gelangt in den Status „möglicher Punkt“, sobald der Benutzer sich diesem in einem Radius von 75 Metern nähert (vgl. **Abbildung 22**). Ab diesem Zeitpunkt wandert der KP in die Liste der möglichen Punkte und wartet darauf, ob er als „Nächster Punkt“ in Frage kommt oder wieder aus dieser Liste entfernt wird (Abbruchbedingungen wurden weiter oben diskutiert). Wenn der Benutzer sich innerhalb des kleineren Kreises mit Radius **A** um den KP bewegt, wird der KP als nächster Punkt markiert. Sobald der Radius **B** verlassen wird, wird der Zustand „Verfolgung“ aktiviert, da diese KPs ebenfalls in Frage kommen können (vgl. **Abbildung 25**). Der KP, welcher die Benachrichtigung sendet, berechnet die Abbruchbedingungen für alle seine Kanten und benachrichtigt alle KPs, welche durch diese Kanten mit diesem KP verbunden sind. Danach setzt sich der KP auf den initialen Status zurück.

Alle Ziel-KP (vgl. **Abbildung 24**) verfolgen, ob sie vom Benutzer anvisiert werden oder nicht. Sobald der Benutzer nahe genug an eines der Ziel-KPs kommt, wechselt der Ziel-KP in den Status „Überprüfung, ob angehalten wurde“. Dabei wurde bei der Entwicklung darauf geachtet, dass es einen sogenannten Näherungsfaktor gibt, welcher viermal so groß ist wie der Radius **A**, um zu bestimmen, ob der Benutzer in der Nähe eines KP ist. Der Grund hierfür ist die Geschwindigkeit, mit welcher sich der Benutzer dem KP nähert. Es wurde festgestellt, dass bei einem Radius von 50 Meter die Wahrscheinlichkeit sehr gering ist bei fahrendem *ÖPNV* sich in der Nähe befindliche KP richtig festzustellen. Durch ausprobieren ergab sich ein Radius von 200 Metern, welcher es möglich macht, bei hohen Geschwindigkeiten noch die sich in der Nähe befindlichen KPs zu erkennen.

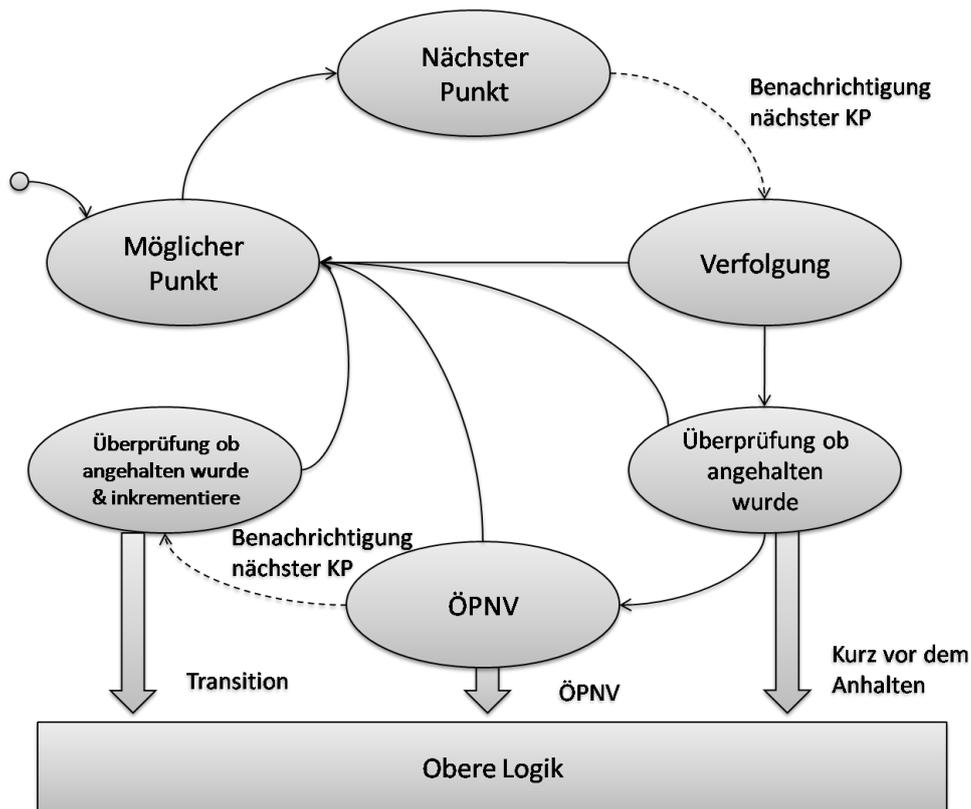


Abbildung 25: Zustandsautomat der unteren Logik

Wenn nun festgestellt wird, dass der Benutzer langsamer wird und stehen bleibt, sendet die untere Logik die Benachrichtigung „ÖPNV“ an die obere Logik. Dank des wissensbasierten Systems, also des Abgleichs der ÖPNV-Infrastruktur mit Hilfe eines gerichteten Graphen, weiß der Algorithmus über den KP Bescheid und kann eingrenzen ob der KP zu einem Bus, einer Tram, einer S-Bahn oder U-Bahn gehört. Wenn beispielsweise dieser KP ausschließlich eine reine Bushaltestelle darstellt, liefert der Algorithmus nach Abgleich mit dem Fahrplan, in welchem Bus sich der Benutzer gerade befindet. Bei einem KP, von welchem mehrere ÖPNV-Transportmittel weggehen, üblicherweise an Hauptbahnhöfen, hat dieser Algorithmus seine Schwächen. Der Algorithmus wird in diesem Falle solange sämtliche in Frage kommenden ÖPNV-Transportmittel anzeigen, absteigend nach ihrer Wahrscheinlichkeit, wie diese anhand der KPs identifiziert sind. Die Erkennungsrate konnte aber deutlich verbessert werden, wenn auch hier mit Hilfe von Echtzeitfahrplänen ein Mapping gemacht wird.

Um nun festzustellen, ob der Benutzer aus einem ÖPNV-Transportmittel heraus und in ein anderes ÖPNV-Transportmittel einsteigt, wird noch der zusätzliche Zustand „Überprüfung, ob angehalten wurde & inkrementiere“ eingeführt, um zu vermeiden, dass die Logik jedes Mal eine „ÖPNV“-Benachrichtigung an die obere Logik sendet. Aus dem Status „ÖPNV“ der unteren Logik benachrichtigt der KP seine nächsten KP und lauscht wieder, ob der Benutzer langsamer wird und anhält. Wenn dies der Fall ist, sendet die untere Logik die Information „Transition“ an die obere Logik. Durch die Hinzunahme sämtlicher sich in der Nähe befindlichen KP kennt der aktuelle KP die Möglichkeiten zur Transition, also der sich in der Nähe befindlichen Haltestellen.

Ab diesem Moment läuft die Überprüfung wieder weiter, indem weiterverfolgt wird, ob sich der Benutzer einem der KPs nähert (A- und B-Radien aus **Abbildung 22**), oder eine der drei eingeführten Abbruchkonditionen tritt ein und es wird dabei entweder die Aktivität „zu Fuß gehen“ oder „Auto“ mittels der *Activity Recognition* von Google festgestellt. **Abbildung 26** zeigt, wie die zwei Algorithmen der oberen und unteren Logik in Zusammenhang stehen.

Die Untere Logik dient der Identifizierung des Zustandes ÖPNV. Die Erkennung der Zustände *Gehen* und *Auto* wird von der *Activity Recognition* von Google übernommen.

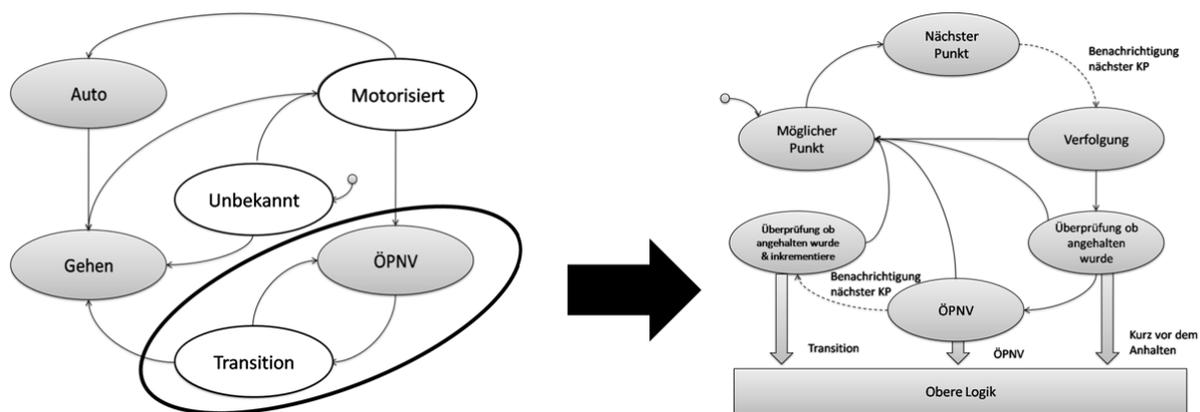


Abbildung 26: Zusammenhang zwischen Oberer und Unterer Logik

3.1.3 Weiterführende Konzepte

In diesem Absatz soll die Frage nach Möglichkeiten, den Akkuverbrauch während der Benutzung des TME-Algorithmus zu reduzieren, diskutiert werden. Bei der Durchführung der Studie, welche im Kapitel 6 beschrieben wird, war einer der größten Kritikpunkte an den entwickelten Prototypen der hohe Akkuverbrauch. Bei der Entwicklung der TME-Applikation wurde der Fokus jedoch auf die während der Literaturrecherche identifizierte Forschungslücke gelegt, nämlich die korrekte Erkennung der verschiedenen motorisierten Zustände (Auto, Bus, Tram, S-Bahn, Zug, etc.) in Echtzeit. Da der KP-Ansatz dafür zwischen einer und fünf Sekunden die GPS-Positionen benötigt, um diese logisch mit der unteren Logik auf mögliche ÖPNV-KPs abzugleichen, ist der Akkuverbrauch sehr hoch, die sehr hohe Erkennungsrate von knapp 97% hingegen aber auch mit den vorhandenen Studien nicht zu vergleichen (vgl. Kapitel 2.2). Bei der Entwicklung dieses Ansatzes war der Akkuverbrauch nebensächlich. Nachfolgend werden zwei Möglichkeiten aufgezeigt, wie die entwickelte Lösung angepasst werden kann, damit der Akkuverbrauch deutlich sinkt.

3.1.3.1 REDUZIERUNG DER ABTAstrate

Bei dem hier entwickelten Algorithmus wurde eine sehr hohe GPS-Abtastrate von einer bis fünf Sekunden umgesetzt. Abhängig von dem aktuellen Status, beispielsweise Auto, kann diese Abtastrate signifikant reduziert werden, da sich während einer Autofahrt der Zustand nicht innerhalb einer bis fünf Sekunden ändert. In Anbetracht des sehr hohen Akkuverbrauchs bei dieser Abtastfrequenz, wurde die Reduzierung dieser Rate

diskutiert. Dazu wurde die Annahme getroffen, dass drei bis fünf Stationen in einer Stadt innerhalb von fünf Minuten erreicht werden können. Die Abstände innerhalb dieser Stationen wären daher sehr kurz, ca. 800 bis 1500 Meter. Die Gefahr, die bei dem entwickelten Algorithmus besteht, ist, relevante KPs nicht rechtzeitig zu erkennen.

Aus diesem Grund, müssen folgende Anpassungen am Algorithmus erfolgen:

Die Abbruchbedingungen müssen angepasst werden.

Eine rekursive Graphensuche müsste eingeführt werden.

In der KP-Logik müssten die Näherungsfaktoren von 50 bis 75 Metern (resp. 200 bis 300 Metern) angepasst werden.

Die **Abbildung 27** zeigt ein Beispiel für eine zwei-Punkte Messung innerhalb einer Stadt. Dabei kennzeichnet die 1 den Zeitpunkt und Ort der ersten Messung und die 2 analog den der zweiten Messung. In Wirklichkeit wäre die KP-Dichte jedoch viel höher. In diesem Beispiel müsste der Algorithmus angeben, dass der Benutzer die Linie 16 gefahren hat, weil die anderen Optionen, 19 und 50 früher abzweigen. Es existiert noch die Möglichkeit, dass der Benutzer auf die Linie 48 gewechselt ist, jedoch wäre es realistischer, wenn die Person auf der Linie 16 geblieben ist, weil die Linie 16 zwischen Punkt 1 und 2 durchgängig verläuft. Der Benutzer hätte auf die Linie 48 wechseln können, wenn er mittendrin ausgestiegen wäre, um noch kurz etwas zu besorgen, und dann an den Punkt zurückgekommen wäre, um dann weiter mit Linie 48 zu fahren. Ein weiterer Grund für Linie 48 wäre, wenn auf Grund einer Betriebsstörung Linie 16 nicht bis zum Punkt 2 weitergehen würde. In beiden Fällen hätte das unser entwickelter Algorithmus zum einen anhand der Activity Recognition festgestellt und zum anderen anhand der Echtzeit-Fahrpläne.

Bei einer geringen Abtastrate besteht die Gefahr, bei dem hier entwickelten Algorithmus nicht alle KPs zu erkennen. Das Beispiel aus **Abbildung 27** verdeutlicht dies. Bei einer durchschnittlichen Gehgeschwindigkeit von 5 km/h könnte eine Person in fünf Minuten ca. 420 Meter zu Fuß zurücklegen. Der Radius des Kreises **B** aus **Abbildung 22** würde dann aus ca. 850 Metern bestehen. Es ist dabei unwahrscheinlich, dass der KP-Abdeckungsbereich dabei zu Fuß nach fünf Minuten verlassen wird. Da in Großstädten, wie beispielsweise München, die KP-Dichte sehr hoch ist, würden jedoch innerhalb dieses Bereichs mehrere KP liegen. Die Erkennung würde sich „hölzern“ und „verlangsamt“ anfühlen, da die Anforderung an diesen Algorithmus war, die TME in Echtzeit zu erhalten.

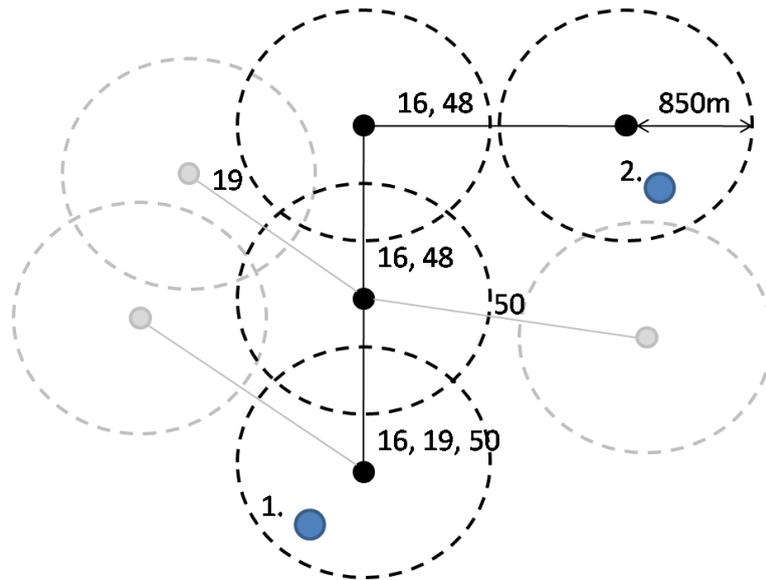


Abbildung 27: Beispiel einer Graphensuche mit niedriger Abtastrate

In weiterführenden Arbeiten könnte ein optimales Verhältnis zwischen KP-Radius, Abtastrate und Akkuverbrauch untersucht werden.

3.1.3.2 KONZEPT DES GEOFENCINGS

Ein Geofence kann mit einem virtuellen Zaun um einen festen Ort verglichen werden. Wenn diese virtuellen Begrenzungen überschritten werden, wird eine Benachrichtigung an eine Applikation gesendet. Mobile Betriebssysteme haben diese Technik bereits implementiert und liefern optimierte und akkusparende Lösungen. Mit Hilfe dieser Lösung kann der hier beschriebene Algorithmus in den Stand-by Modus versetzt werden, sobald der Benutzer sich für eine bestimmte Zeit nicht mehr bewegt. Dabei wird eine Geofence rund um den aktuellen Bereich erstellt. Sobald der Benutzer diesen Bereich wieder verlässt, springt die TME-wieder an und verfolgt die Zustände.

3.1.3.3 AUTOMATISIERTES KP-MAPPING

Der kritischste Teil für den hier entwickelten Algorithmus ist die richtige und vollständige Datenbasis der KP in der benötigten Qualität. Nur dann kann die vorgestellte TME ihre Erkennungsraten von den hier erzielten 95% erreichen. Da der hier entwickelte KP-Ansatz neu ist, gibt es keine Datensätze mit KP für verschiedene Städte. Um sich Abhilfe zu beschaffen, wurden die KPs im ersten Schritte manuell aus der öffentlich zur Verfügung gestellten OSM für München extrahiert. Dabei mussten einzelne Haltestellen und Linien manuell nachgezogen werden, weil sie nicht auf dem aktuellsten Stand waren. Um sich diese Arbeit zu ersparen, wäre eine Möglichkeit, die KPs über sog. Crowdsourcing zu erhalten. Mit anonymisierten Positionsdaten von tausenden von Leuten, kann ein solcher Algorithmus KPs automatisch identifizieren. Indikatoren für KPs wären dabei Orte, welche eine hohe Dichte an Personen zur gleichen Zeit aufweisen. Dabei könnten mit Hilfe des Map-Matching-Verfahrens Restaurants, Kinos, Appartements etc. herausgefiltert werden. Durch diesen Ansatz könnte nicht nur die Information über den aktuellen Ort analysiert werden, sondern auch fehlende Information. Beispielsweise

wären Orte, an denen viele Personen für eine bestimmte Zeit das GPS-Signal verlieren, Indikatoren für U-Bahn-Stationen.

3.2. DATENSAMMLUNG UND –VERARBEITUNG

Der hier beschriebene Algorithmus arbeitet mit der Idee der KPs. In diesem Kapitel soll beschrieben werden, wo diese KP herkommen und wie diese verarbeitet und wie die Datenstruktur dazu aussieht.

Der entwickelte Prototyp wurde für das Smartphone-Betriebssystem Android gebaut. Die weitverbreitetste Datenrepräsentation dabei ist entweder XML oder SQLite. Die Verwendung von XML als Repräsentation für Daten wurde dabei ausgeschlossen, weil es keine schnelle und effiziente Datensuche erlaubt. Aus diesem Grund wurde beschlossen, eine SQLite Datenbank für die KPs zu erstellen. Dabei bietet SQLite keine Möglichkeit räumliche Daten abzubilden. Deshalb wurde die räumliche Datenrepräsentation mittels mehrerer Spalten dargestellt.

3.2.1 Datensammlung

Die meisten Kartenanbieter bieten eine frei zugängliche Schnittstelle an, für den Zugang und die Suche innerhalb ihrer Datenbank. Dazu gehören bekannte Vertreter wie Google Maps oder Bing Maps von Microsoft. Obwohl sie eine offene Schnittstelle besitzen, gibt es keine Möglichkeit an die Rohdaten zu gelangen, welche jedoch für den entwickelten Algorithmus benötigt werden um alle benötigten KPs zu exportieren, welche in Verbindung zum ÖPNV-Netz stehen. Zusätzlich wird der Zugriff auf die entsprechenden Datenbanken schnell blockiert, wenn innerhalb weniger Minuten viele Suchanfragen eingehen. Die alternative Lösung stellt Open Street Map (**OSM**) dar, welches als Datenbasis für diese Arbeit verwendet wurde. Auf der Seite von OSM wird der Zweck so beschrieben: „... Gemeinschaft von Kartographen, welche weltweit Daten zu Straßen, Pfaden, Cafés, Haltestationen, und vieles mehr erheben und pflegen“ (Map, 2014).

Wie viele Open Source Projekte wie beispielsweise Wikipedia, leidet auch OSM unter dem Problem, dass jeder zu jeder Zeit die Daten manipulieren kann. Natürlich werden die Daten von anderen Mitgliedern der Gemeinschaft kontrolliert und Personen, welche diese Open Source Projekte missbrauchen, werden aus der Community ausgeschlossen. Trotzdem unterliegen solche Systeme menschlichen Fehlern und die Datenqualität hängt sehr stark von den Mitgliedern der jeweiligen Region ab. Der Vorteil ist, dass Änderungen an den Daten normalerweise ziemlich schnell umgesetzt werden und die Karten dadurch aktueller sind, als die der meisten Anbieter. So taucht eine neu errichtete Brücke beispielsweise viel früher in solchen Karten auf, als bei Anbietern, wie Google Maps. Einer der größten Nachteile ist jedoch, dass die Daten „chaotisch“, also es zu ein und dem gleichem Datum verschiedene Schreibweisen geben kann, sind und dadurch schwer in der Verarbeitung.

Auf die Rohdaten von OSM kann durch sogenannte *Planet-Dateien* zugegriffen werden. Diese Dateien besitzen ihr eigenes Rohformat, in welchem die Daten komprimiert wurden. Die Datei enthält eine Historie über die Datensätze, welche wichtig für die Open Source Community ist, jedoch für diese Arbeit irrelevant ist. Die Funktionalität der Java Bibliotheken ist für diese Zwecke nicht ausreichend. Aus diesem Grund musste ein anderes Werkzeug dafür verwendet werden, welche die Rohdaten auf eine lokale Datenbank exportiert, welche

in der Lage ist die räumlichen Daten abzuspeichern (osm2pgsql (OSM2PQSQL, 2014)). Die Datenbank, die für diesen Zweck verwendet wurde, ist PostgreSQL (PostgreSQL, 2014). Für die Verwendung von osm2pgsql wird die Verwendung einer Unix-Shell Umgebung empfohlen. Bei dem Exportieren jeder OSM Planet Datei auf eine lokale Datenbank werden folgende vier Haupttabellen generiert:

planet_osm_point

Diese Punkttabelle enthält alle Punkte eines ÖPNVs. Das sind in den meisten Fällen die POI aber auch Punkte wie Sitzbanken, Mülleimer, Hydranten, etc.

planet_osm_lines

Die Linien-Tabelle besteht aus Straßen und allem was sonst als Linie repräsentiert werden kann, wie z.B. Wanderrouten oder ÖPNV-Routen.

planet_osm_polygon

Die Polygonen-Tabelle besteht aus Bereichen. Dazu gehören in erster Linie Seen, Grünflächen, etc. Die Plattformen von Haltestellen sollten ebenfalls Teil dieser Tabelle sein. Aber zur Zeit der Erstellung dieser Arbeit, konnte osm2pgsql nicht mit Multi-Polygonen umgehen und viele Polygone wurden in der Linien-Tabelle gespeichert.

planet_osm_roads

Die Straßentabelle repräsentiert eine höhere Auflösung von manchen Linien, welche in der Linien-Tabelle abgelegt sind.

Osm2pgsql wird hauptsächlich dafür verwendet eine graphische Repräsentation der Roh OSM-Daten zu liefern. Das bedeutet, dass alle logischen Verbindungen während des Exports verloren gehen. Jeder Eintrag, welcher vorher eine Verbindung dargestellt hat wird bei dem Export zu einem Eintrag mit negativem Vorzeichen konvertiert. So wird beispielsweise eine Verbindung, welche in der ursprünglichen OSM-Tabelle die ID 25 hatte, in mehreren Zeilen mit der ID -25 dargestellt. Welche Art von Objekten in einer Relation sich befindet ist nicht weiter spezifiziert. Ein Beispiel für eine Verbindung ist eine U-Bahn Station. Es enthält Punkte, welche mit diesem Bereich in Verbindung stehen, wie die Position von Geldautomaten, Geschäften oder die Haltestellen der Züge. Es ist aber nicht garantiert, dass alle Verbindungen richtig zugeordnet werden. Manche Punkte stehen für sich alleine dar, ohne Verbindung.

Linien werden in der Datenbank mit Hilfe von Tags wiedergefunden. Folgende Tags sind für den ÖPNV wichtig: „subway“, „tram“, „bus“, „train“, „light_rail“, and „monorail“, wobei der Tag „train“ für große Entfernungen verwendet wird und „light_train“ mit S-Bahnen gleichzusetzen ist. Eine Beispielabfrage könnte so aussehen:

```
SELECT * FROM planet_osm_lines where route = 'tram';
```

Wenn man nun die Ergebnisse dieser Abfrage analysiert, fällt auf, dass Linien zwischen mehreren Segmenten aufgeteilt werden. Manche Linien bestehen aus über 20 Segmenten. Aufgrund des Multi-Polygonen-Fehlers liefert diese Abfrage viele Plattformen, welche eigentlich herausgefiltert werden sollten.

Folgender Ablauf für den Export der Daten in die Datenbank wurde verfolgt:

Exportiere Stationen und markiere diese mit dem Tag (Bus, Tram, Zug, etc.)

Exportiere die ÖPNV-Routen.

Versuche Routen zu kombinieren, welche den gleichen Namen haben

Sortiere kurze und geschlossene Pfade aus. Dies ist ein Workaround um die Plattformen zu eliminieren, welche nicht Teil dieser Liste sein dürfen.

Finde die Richtung, in welche die Linie fährt, aus dem „From“- und „To“-Tag extrahiert. Wenn nicht verfügbar, analysiere das Beschreibungsfeld

Folge der kombinierten ÖPNV-Route und füge Stationen hinzu, welche sich in der Nähe der Route befinden und den richtigen Transportmittel-Typ besitzen.

Stelle einen logischen Graphen zwischen allen Stationen/KPs auf.

Finde Transitionen zwischen multiplen KPs.

(Version 1)

Dieses Vorgehen hat für das ÖPNV-Netz in München sehr gut funktioniert und es wurden sehr viele Daten produziert. Manche Stationen mit fehlenden Namen wurden manuell zur OSM-Datenbank hinzugefügt.

Planet_osm_rels

Diese Tabelle enthält die Beziehungen, wobei jede Beziehung eine Liste an IDs enthält. Jede ID kann entweder ein Knoten ('n'), ein Weg ('w') oder eine andere Verbindung ('r') sein.

Planet_osm_nodes

Enthält die Knoten. Diese Tabelle ist vergleichbar mit der Tabelle planet_osm_lines.

Planet_osm_ways

Enthält Linien, Polygone und Multi-Polygone. Diese werden in planet_osm_lines und planet_osm_polygon repräsentiert.

Mit Hilfe der planet_osm_rels Tabelle und den IDs welche bereits bekannt waren, konnte ein zuverlässiger Tag, route_masters, in der Relationen Tabelle gefunden werden. Diese route_masters enthalten die meiste Zeit zwei Relationen, eine für jede Richtung welche die ÖPNV-Linie fahren kann. Manche route_master Einträge enthalten mehr als zwei Verbindungen, und manche gar keine sondern nur eine Liste von Knoten und Wegen. Der Vorteil von den Verbindungen ist, dass die Punkte präziser sind. Vorher mussten diese zu dem nächst gelegenen Punkt mit dem korrekten Namen gemappt werden. Gleichzeitig besitzen die meisten Punkte keinen Namen. Mittels einer rekursiven Suche durch die Relationen welche dieser Punkte zugewiesen haben, war es möglich die korrekten Namen zu den unbekannt Positionen zuzuweisen. Die letzten noch fehlenden Namen wurden durch die Namen der sich in der Nähe befindlichen Stationen ergänzt.

Der neue Ablauf sieht nun folgendermaßen aus:

Exportiere die route_master Relationen und analysiere alle Verbindungen. Kombiniere die Routensegmente und füge die referenzierten Stationen hinzu.

Ordne die Namen mittels rekursiver Relationen Suche zu und nachträglich mittels der Suche nach dem nächsten Nachbar.

Sortiere die Stationen nach den Positionen innerhalb der kompletten Route.

Erstelle einen logischen Graphen zwischen allen Stationen/ KP.

Finde Transitionen zwischen multiplen KP.

(Version 2)

Der zweite Ansatz (Version 2) hatte mehr Probleme bei der Erkennung der Routen von Bussen als er erste Ansatz (Version 1), weil viele Bus-master_relations nicht richtig dargestellt wurden. Der zweite Ansatz richtet sich aber mehr nach der OSM Spezifikation und wird dadurch in Zukunft weiter verwendet werden können.

3.2.2 Datenmodellierung und digitale Karte

Das ER-Datenmodell, welches dem Algorithmus zur TME zugrunde liegt, ist in **Abbildung 28** dargestellt. Es repräsentiert die Daten von Stationen, Linien, die Verbindungen zwischen den Stationen, die Kanten, die KP-Cluster und die Transitionenpunkte. Nachfolgend werden die wichtigsten Tabellen beschrieben. **Tabelle 4** stellt die Haupttabelle für die Repräsentation der Stationen-KP dar. Dabei handelt es sich um eine Untertabelle der Tabelle „Keypoint“, aber anstelle eines Fremdschlüssels zur Repräsentation der Keypoint-Tabelle, wurden alle Parameter vererbt. Aus Gründen der Performanzoptimierung, besitzen die Parameter von Latitude und Longitude ihren eigenen Index.

Station	
Name	Datentyp
id	int (pk)
idType	int (fk)
idKPcluster	int (fk)
name	text
gps_lat	real
gps_long	real
description	text
wheelchair	boolean

Tabelle 4: SQLite-Tabelle zum Speichern von Stationen-KPs

Tabelle 5 zeigt die SQL-Tabelle für die Linien. Linien mit demselben Namen aber unterschiedlicher Richtung bekommen mehrere Tabelleneinträge.

Line

Name	Datentyp
id	int (pk)
idType	int (fk)
name	text
from	text
to	text

Tabelle 5: SQLite-Tabelle für Linien

Für die Repräsentation des Graphen für die Station, wird **Tabelle 6** verwendet. Es stellt die logische Verbindung zwischen multiplen KP dar und ist dadurch eine der wichtigsten Tabellen für die Entscheidungsfindung. Der gerichtete Graph wird dabei einfach dargestellt, indem der Quelle-KP auf den Ziel-KP zeigt und Bezug zu einer Kante nimmt. Die Kante enthält eine Liste von mehreren Linien, welche in **Tabelle 7** repräsentiert werden.

StationGraph

Name	Datentyp
idStation	int (fk)
idTargetStation	int (fk)
idEdge	int (fk)

Tabelle 6: SQLite-Tabelle zum Speichern des Graphen zwischen den KPs.

EdgesToLines

Name	Datentyp
idEdge	Int (fk)
idLine	Int (fk)

Tabelle 7: SQLite-Tabelle beinhaltet die Kanten und die entsprechenden Linien aus dem ÖPNV

Die **Tabelle 8** wird verwendet um KP weiter zu definieren und den Umgang mit ihnen zu beschreiben. Die Typen der KP geben Auskunft darüber, wie mit den KP umzugehen ist. Definierte KPs sind SBahn, Bahn, Bus, Tram, Subway, Monorail, KPEntrance, KPElevator, KPStop, KPParkAndRide, Unsupported.

Type

Name	Datentyp
id	int (pk)
name	text

Tabelle 8: SQLite Tabelle speichert verschiedene Typen um die KPs detaillierter beschreiben zu können.

Auf die Datenbank kann mittels SQL-Abfragen zugegriffen werden. Nachfolgend sind zwei Beispiele genannt, wie ein solcher Zugriff auf die Daten aussehen könnte. Das erste Beispiel liefert einen KP, welcher sich innerhalb eines bestimmten Bereichs befinden. Diese Information benötigt der Algorithmus um alle KPs die sich in der Nähe eines KPs befinden zu finden.

```
SELECT S.id , S.idKeypointCluster , S.name , S.description , S.gps_lon,  
       S.gps_lat, S.idType FROM Station as S WHERE S.gps_lat BETWEEN ? AND ? AND  
S.gps_lon BETWEEN ? AND ?;
```

Die zweite Abfrage ist etwas komplizierter. Dieser Abruf liefert eine Liste aller möglichen Linien zurück, welche an einem bestimmten KP abfahren.

```
SELECT L.id , L.idType , L.name , L.FromDirection , L.ToDirection FROM  
       StationGraph AS SG CROSS JOIN EdgesToLines AS EtL CROSS JOIN Line AS L WHERE  
       SG.idStation = ? AND SG.idEdge = EtL.idEdge AND EtL.idLine = L.id
```

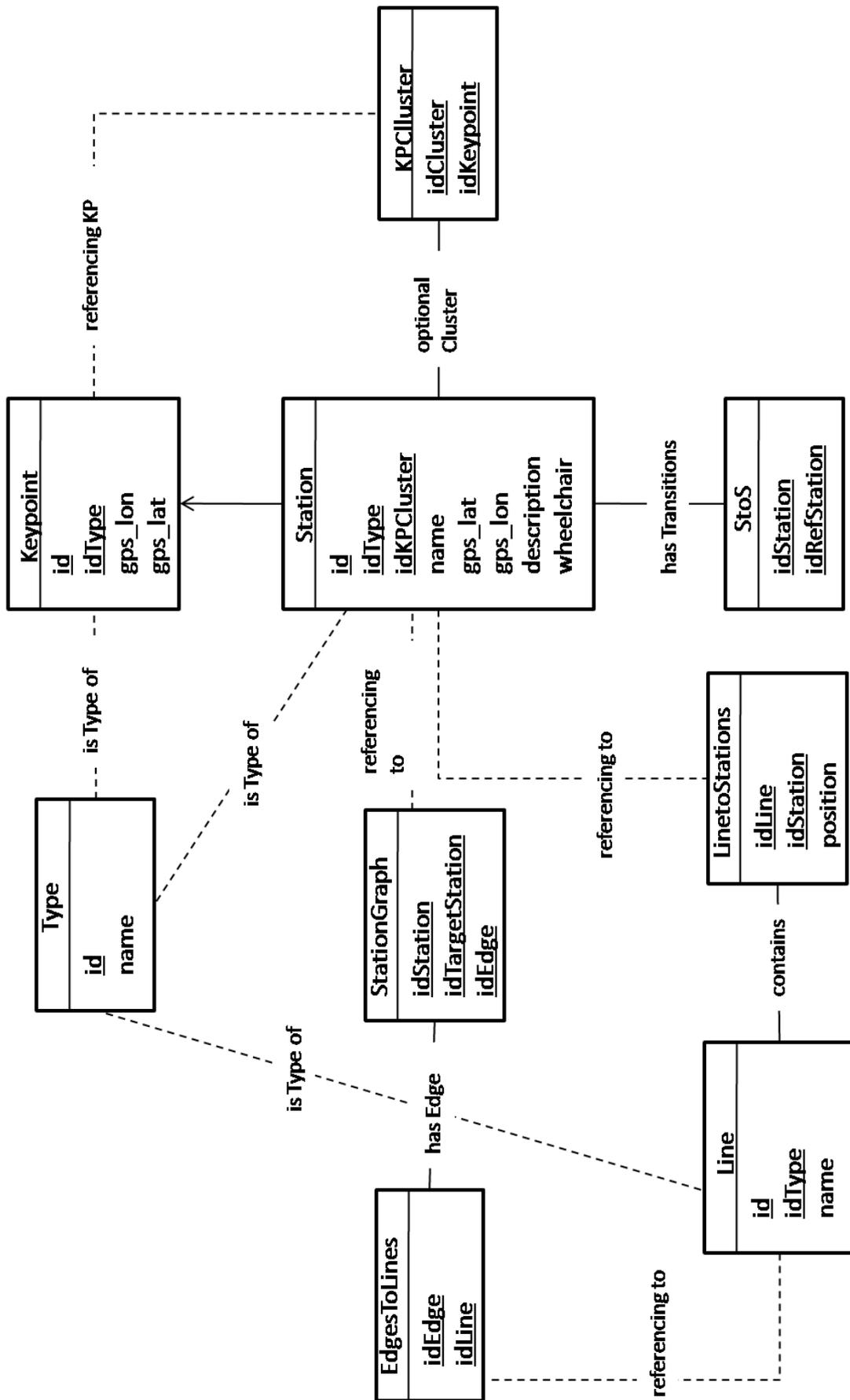


Abbildung 28: Abstraktes Datenmodell zur Repräsentation von KPs, Stationen und Linien

3.3 CASE STUDY UND AUSWERTUNG

In diesem Kapitel wird die Auswertung des Algorithmus anhand des entwickelten Prototypen diskutiert. Das Ziel des Prototypen war es, mit Hilfe einer einfachen wahr oder falsch Oberfläche die korrekte Erkennung des Algorithmus zu überprüfen. Die Fallstudie wurde in München durchgeführt. Dabei wurden 3.000 KPs, 4.000 Kanten und ca. 360 Linien vom Exporter geliefert. Der Prototypen-Test dauerte vier Tage und wurde von elf Personen durchgeführt. Im Zuge der weiteren Forschung zu dieser vorliegenden Arbeit wurden zwei größere Studien durchgeführt, welche neben der Funktionsweise auch die Akzeptanz des Prototypen untersucht haben. Der Studienaufbau, die Teilnehmerzahl und die Ergebnisse dazu sind in Kapitel 6 dieser Arbeit zu finden.

3.3.1 Prototyp

Wie beschrieben, war das Ziel dieses Prototypentests die korrekte Erkennung der einzelnen Transportmittel. Deshalb war der Anspruch an den Prototypen eine einfache Oberfläche zu besitzen, auf welcher der Benutzer bei korrekter Erkennung des Transportmittels dieses bestätigen kann und bei einer Nicht-korrekten Erkennung dieses melden kann. Im späteren Verlauf der Forschungsarbeit wurde ein aufwändiger Prototyp konstruiert, welcher eine komplette intermodale Navigation erlaubt. Die Testergebnisse sind sehr vielversprechend. Die unbereinigte Erkennungsrate des korrekten Transportmittels im ÖPNV lag bei 90,9% und die bereinigte bei 96,77%. Bei der bereinigten Zahl wurden falsche Erkennungen aufgrund von verlorenem GPS-Signal, wenn sich die Person unterirdisch aufgehalten hat und aufgrund von graphischen Problemen herausgefiltert. Graphische Probleme traten im Zentrum der Stadt auf, bei der sehr viele Vorschläge für das aktuelle Verkehrsmittel geliefert wurden und der richtige Vorschlag aufgrund der Größe des Anzeigebildschirms nicht auftauchte.

Die elf Tester haben den Prototypen auf ihren Android Smartphones jeden Tag, vier Tage insgesamt, auf dem Weg in- und aus der Arbeit sowie, immer wenn sie unterwegs waren, verwendet. Der Fokus dieser Fallstudie lag dabei nicht auf der Quantität, sondern auf der Qualität um die Funktionalität des entwickelten Algorithmus zu testen. Die elf Personen wurden so ausgewählt, dass sie sowohl öffentlich als auch mit dem Auto fahren werden und dass sie möglichst an unterschiedlichen Orten in München unterwegs sein werden.

Sobald die Benutzer unterwegs waren konnten sie mit dem Button „Activate Service“ die TME starten und anonym Daten senden. Mit dem Button „Deactivate Service“ wurde die Anwendung wieder beendet. Die anonymisierten Ergebnisse enthielten einen Zeitstempel, die gelieferte Antwort, den Ort an welchem die Erkennung gemacht wurde und einen Kommentar, falls dieser hinzugefügt wurde. Sobald die TME- Applikation eine Transportart identifiziert hatte, wurde der Benutzer darüber benachrichtigt. Daraufhin konnte der Benutzer eingeben, ob die Erkennung richtig war (TRUE) oder falsch (FALSE). Im Falsch-Fall, öffnete sich ein Kommentarfeld, in welches der Benutzer eine Begründung eintragen konnte. Ein Kommentar konnte nur dann hinzugefügt werden, wenn die Erkennung falsch war. Um die Auswertung zu vereinfachen, enthielt der Kommentarbildschirm vordefinierte Antwortmöglichkeiten.

3.3.2 Typisches Szenario

Eingangs in Kapitel 3.1.2 wurden Problemklassen definiert, welche mit dem TME-Algorithmus erkannt und behoben werden sollen. Die Problemklassen zeigen alle Möglichkeiten auf, welche ein Benutzer hat um von

seinem Start zu seinem Endpunkt zu gelangen unter Verwendung der Transportmittel *Auto*, *ÖPNV* und *Gehen*. Das Szenario soll nun die Problemklassen B (Gehen → ÖPNV) und D (Umsteigen) behandeln und die Funktionsweise des Algorithmus und somit der TME belegen. **Abbildung 29** zeigt das angesprochene Szenario. Dabei startet der Benutzer im Nord-Osten von München, an der Station Fritz-Meyer-Weg. Auf dem Weg zur Station wurde zunächst der Modus *Gehen* von der Applikation richtig erkannt. Wie in Kapitel 3.1.2.1 beschrieben ist, läuft diese Erkennung über die *Activity Recognition* von Google. An dieser Station gibt es insgesamt drei mögliche Linien (Bus 50, Tram 16, 18), in Summe also sechs Möglichkeiten. Bei diesem Szenario wurde die Tram 16 in Richtung Innenstadt gewählt. Die drei Möglichkeiten stadtauswärts wurden nach dem Losfahren von der Applikation ziemlich schnell aussortiert und nicht weiter betrachtet. Sobald die nächste Station erreicht wurde, zeigt der Algorithmus alle drei Linien an diesem Punkt korrekt an. Die Erkennung der Tram 16 verbesserte sich nach ein paar weiteren Stationen, als die anderen Linien weiter abzweigten. Wie man an diesem Beispiel sieht, hätte die Erkennung viel früher die richtige Linie liefern können, wenn die Datenbank der MVG mit ihren Live-Abfahrtszeiten zum Abgleich hinzugenommen wäre. Während dieser Fahrt wurden alle Stationen, welche die Tram Nr. 16 angefahren hat, auf dem Smartphone in dem Moment angezeigt, in dem die Tram diese Haltestellen erreichte. Das Ziel dieser Route war die Station Max-Weber-Platz, an welcher ein Umstieg zur Tram 19 vorgenommen wurde.



Abbildung 29: Routenverlauf des Testlaufs

In **Abbildung 29** sieht man im unteren Teil den Umsteigepunkt. Die jeweiligen weiteren Möglichkeiten befinden sich alle auf einem sehr kleinen Raum. Zunächst hat der Algorithmus den Ausstieg aus der Tram nicht erkannt, weil die Bewegung zu dem Umsteigepunkt (Transitionen-KP) minimal ist. Nach ungefähr ein bis zwei Minuten wechselte der Status von *ÖPNV* zu *Gehen*, höchstwahrscheinlich aufgrund des Timeouts für den ursprünglichen Transportmodus (s. Kapitel 3.1.2.1). Am Max-Weber-Platz gibt es auf einem sehr kleinen Raum ungefähr 18 Möglichkeiten für den öffentlichen Nahverkehr. Obwohl dies eine große Herausforderung darstellt, hat der

entwickelte Algorithmus die Transition zu einem neuen ÖPNV-Transportmittel erkannt. Innerhalb einer kurzen Zeit wurden die anderen Möglichkeiten eliminiert und die TME wurde korrekt auf die Tram der Linie 19 gesetzt.

In der Nähe mancher KPs, welche sich in einer Distanz kürzer als 600 Meter voneinander befanden, erkannte der Algorithmus fälschlicherweise, dass die Tram verlassen wurde. Die Logik hatte sich nach kurzer Zeit regeneriert und erkannte wieder richtig die Tram 19. Der finale Ausstieg der Route an der Station München Hauptbahnhof, wurde ebenfalls korrekt erkannt.

Bei diesem Szenario wurden die Umweltgeräusche, wie die Haltestellenansagen sowie die Tramgeräusche während der Fahrt, aufgezeichnet. Somit kann dieses Szenario mit Hilfe dieser Audio-Datei jederzeit nachvollzogen werden. Dabei wird die Audio-Datei mit den GPS-Logs und einem Live-Bild der TME abgespielt. So sieht man dann, wenn die Tram-Ansage für die Station kommt, dass diese auch in dem Prototypen (Live-Bild) angezeigt wird.

3.4 ZUSAMMENFASSUNG

Das Ziel einer TME ist es, das aktuell verwendete Transportmittel (*Gehen, Auto, ÖPNV*) automatisiert und in Echtzeit korrekt zu identifizieren. Dazu wurde ein Ansatz entwickelt, welcher auf sogenannten Key-Points aufsetzt. Die Key-Points stellen dabei charakteristische Punkte, wie Haltestellen im ÖPNV, dar. Der Algorithmus verwendet für die TME zum einen die *Activity Recognition* von Google für die Erkennung der Modi *Gehen* und *Auto* aber auch um Abbruchbedingungen zu erkennen. Für die Differenzierung zwischen den einzelnen ÖPNV-Zuständen verwendet der TME-Algorithmus das in dieser Arbeit entwickelte Key-Point Konstrukt. Der Anspruch an den Algorithmus war es weiterhin, auf Android-Smartphones zu funktionieren, da in der heutigen Welt, Smartphones ständige Begleiter der Menschen darstellen und somit bei allen Transportmitteln, auch beim Wechsel, immer dabei sind. Bei der Literaturrecherche kristallisierten sich dabei zwei Ansätze heraus, wie mit Hilfe von Smartphone-Sensoren TME-Algorithmen aussehen können. Dabei wird im ersten Ansatz ein Reasoning mit Hilfe von maschinellem Lernen der Logs der Smartphone-Sensoren vollzogen. Der zweite Ansatz ist es logische Regeln zu erstellen auf deren Basis die TME funktionieren soll. Beide Ansätze liefern in der Literatur gute Ergebnisse. Da die Idee des Key-Points-Konstrukts ziemlich früh entstand, wurde der regelbasierte Ansatz gewählt, welches mit Hilfe von Smartphone-Sensordaten und Umweltinformationen (**GIS**) eine Erkennungswahrscheinlichkeit des richtigen Transportmittels von 96,77% liefert. Dabei ist hier wichtig zu erwähnen, dass die Erkennung in Echtzeit läuft. Viele Publikationen liefern zwar auch gute Erkennungswahrscheinlichkeiten, jedoch meistens basierend auf offline-Daten. Zusätzlich ist hier zu erwähnen, dass bei diesem Ansatz, die einzelnen Segmente (Bsp. Fußweg, Auto, ÖPNV, Fußweg, ÖPNV, Fußweg) automatisiert vom Algorithmus erkannt werden. In der Literatur funktionieren die meisten Ansätze nicht automatisiert, sondern erfordern manuellen Input vom Benutzer. Neben der automatisierten Echtzeiterkennung war der Anspruch an diesen TME-Algorithmus bessere Erkennungsraten zu liefern, als die Methoden in der Literatur, also mind. 93%, für die ÖPNV-Erkennung mind. 85%. Schließlich überzeugt die sehr hohe Erkennungswahrscheinlichkeit von über 95%.

Wie alle Ansätze welche auf gute Datenqualität angewiesen sind, können falsche Daten in der Datenbank zu falschen Entscheidungen des Algorithmus führen. Eine falsche TME war immer auf falsche oder fehlende Daten zurückzuführen, nicht jedoch auf die Funktionsweise selbst. Damit dieser Ansatz jedoch weiter verfolgt werden kann, müssen Maßnahmen zur sehr besseren Datenqualität erfolgen. Weiterhin kann der Algorithmus dahingehend verbessert werden, als dass der Akkuverbrauch mittels Geofencing oder einer geringeren GPS-Abtastrate verringert werden kann. Der Algorithmus funktioniert nur dann zuverlässig, wenn die Annahme, dass die Transportmittel an allen Stationen halten, zutrifft. In zukünftigen Arbeiten kann sich der Fragestellung gewidmet werden, wie hoch die Erkennung des Algorithmus ist, wenn das ÖPNV nicht an allen Stationen hält.

4 PROGNOSE VON DEVIATIONEN IM INTNAV

In diesem Kapitel wird ein Konzept präsentiert, mit welchem Abweichungen innerhalb intNAVs möglichst frühzeitig prognostiziert und hinsichtlich beabsichtigter und unbeabsichtigter Deviation (Anomalien) bewertet werden können. Dabei werden zunächst sämtliche mögliche Situationen innerhalb einer intermodalen Route aufgezeigt und dabei mögliche Anomalien dargestellt (1). Aufbauend darauf werden mögliche Arten von Abweichungen innerhalb (intermodaler) Routen abgeleitet (2). Beginnend mit den Hypothesen (3) wird das Konzept für die Prognosen von Anomalien basierend auf drei Dimensionen (Raum, Zeit und Logik) in intNAVs präsentiert (4), wobei die Berechnung der Abweichung in der Logik ein Novum darstellt. Im letzten Abschnitt (5) werden die Ergebnisse zusammengefasst.

Die **Hauptaufgabe** eines intNAVs besteht darin dem Benutzer in jeder navigatorischen **Situation** die **relevante Information** über seine Route zur **richtigen Zeit** und in der **passenden Form** auf seinem Smartphone zur Verfügung zu stellen. Die Situation beeinflusst also die Information welche dem Benutzer präsentiert werden soll und die Art wie diese Information dem Benutzer präsentiert wird. Innerhalb jeder Situation können Abweichungen entstehen. Da die zur Verfügung stehende Fläche auf einem Smartphone begrenzt ist, ist es sinnvoll dem Benutzer nur die relevante Information zur richtigen Zeit zur Verfügung zu stellen und ihn nicht zu sehr mit unnötigen Informationen zu nerven. Bei Abweichungen von der Route kann der Benutzer dabei absichtlich oder unabsichtlich handeln. Bei traditionellen Navigationssystemen wird keine Unterscheidung zwischen der Ursache für Abweichungen gemacht, sodass bei einer erkannten Abweichung der Benutzer immer darauf hingewiesen wird. Im Rahmen der vorliegenden Forschungsarbeit soll jedoch untersucht werden, wie beabsichtigte von unbeabsichtigten Abweichungen erkannt werden können, sodass je nach Art eine andere Form der Benachrichtigung auf dem Smartphone des Benutzers erscheint. Zur Zeit der vorliegenden Forschungsarbeit gibt es keine vergleichbare Arbeit, welche sich damit beschäftigt, Ursachen für Abweichungen in Navigationssystemen aufzudecken. Die hier vorgestellte Technik ist nicht auf intNAVs beschränkt und kann auch auf traditionelle Navigationssysteme angewendet werden um dem Benutzer noch besser bei seiner Routenführung zu unterstützen und diesen nicht mit unnötigen Informationen zu nerven.

Wenn der Benutzer absichtlich von der Route abgewichen ist, wäre eine Information über die Abweichung für den Benutzer wenig wert, da dieser sich darüber im Klaren ist. Es besteht sogar die Gefahr den Benutzer durch

penetrantes Hinweisen auf die Abweichung zu nerven, wenn dieser die Abweichung absichtlich herbeigeführt hat. Bei unbeabsichtigten Abweichungen würde eine Information darüber einen Mehrwert für den Benutzer darstellen. Bei Smartphones, bei welchen die Fläche zur Darstellung von Informationen begrenzt ist, ist eine Identifikation der Ursache der Abweichung nützlich. Dadurch kann sichergestellt werden, dass der Benutzer nur hilfreiche Informationen bekommt.

Die Situation zu verstehen, in welcher sich der Benutzer aktuell befindet, ist also ein wesentlicher Bestandteil eines Navigationssystems. Nachfolgend werden jedoch nur Situationen betrachtet, welche für ein intNAV relevant sind. Aus diesem Grund wird ein Beispielszenario diskutiert, welches mögliche intermodale Alltagssituationen beschreibt, die ein Benutzer bei der Verwendung eines intermodalen Navigationssystems erleben kann.

Der Fokus des vorhergehenden Kapitels 3 lag darauf, das vom Benutzer aktuell verwendete intermodale Transportmittel *automatisiert* zu erkennen. Das ist der erste grundlegende Pfeiler eines intNAVs (vgl. Kapitel 2). Der zweite grundlegende Pfeiler ist es, festzustellen ob es sich dabei auch um das geplante Transportmittel (wie vom intNAV vorgeschlagen) handelt. Im Falle einer Abweichung (Deviation) soll der Benutzer, ähnlich wie bei einem AutoNAV, eine Meldung bekommen, dass eine Abweichung vorliegt und intermodale Alternativen sollen dem Benutzer dabei präsentiert werden.

Deviationen lassen sich dabei, je nach *Ursache* und *Wirkung* (= **Kontext einer Deviation**), klassifizieren (Kapitel 4.2). Dieses Kontextwissen hat Auswirkungen auf die *Form*, wie die Information einer Abweichung, dem Benutzer kommuniziert werden soll.

Nehmen wir als Beispiel einen Benutzer, welcher einer vorgeschlagenen Route auf seinem intermodalen Navigationssystem folgt. Der Benutzer ist auf dem Weg zum Einkaufen und somit nicht unter Zeitdruck, da danach keine weiteren Termine anstehen. Während der Benutzer, wie vom intNAV empfohlen, an einer Haltestelle umsteigen möchte, sieht dieser eine Postfiliale, die er aufsucht. Da dies vorher nicht geplant war und somit auch nicht vom intNAV einberechnet wurde, stellt dieses eine Abweichung von der geplanten Route fest. Dabei fängt es an den Benutzer darüber in Kenntnis zu setzen und ihm sämtliche Alternativen aufzuzeigen, wie dieser doch noch, ohne großen Zeitverlust, sein vorher ins intNAV eingegebene Ziel erreichen kann. Da der Benutzer jedoch diese Abweichung *absichtlich* gewählt hat, ist die Information darüber, dass eine Abweichung vorliegt aus Sicht des Benutzers, *nutzlos*. Zusätzlich wird es den Benutzer sehr schnell nerven, wenn das intNAV ihn penetrant daran erinnert, so schnell wie möglich seine Route weiterzuverfolgen.

Anders hingegen sieht es aus, wenn der Benutzer auf dem Weg zu einem Termin ist und eine Abweichung nicht absichtlich herbeigeführt hat, sondern diese durch ein Transportmittel initiiert wurde. In solchen Fällen möchte der Benutzer sehr wohl möglichst frühzeitig und vermutlich penetrant vom intNAV darüber in Kenntnis gesetzt werden, dass eine Abweichung vorliegt aber vielmehr, welche Alternativen für den weiteren Routenverlauf bestehen, damit das Ziel ohne großen Zeitverlust erreicht werden kann.

Die Erkennung des Kontextes von Abweichungen, also deren Ursache und die Wirkung auf die nachfolgende Route, in Hinblick auf die Kommunikation mit dem Benutzer, stellt aktuell eine Forschungslücke dar. Aktuelle Arbeiten zur Erkennung von Deviationen in (intermodalen) Routen beschäftigen sich vornehmlich damit, Abweichungen möglichst präzise und möglichst früh zu erkennen sowie die Berechnung von Alternativen zu verbessern. Jedoch kann das Wissen über den Kontext einer Abweichung nicht nur dazu beitragen, den Benutzer in unwichtigen Situationen nicht zu nerven und in wichtigen Situationen eine Hilfe zu sein, sie hilft auch dabei, die Verwendung eines intNAVs möglichst effizient zu gestalten und den verfügbaren knapp bemessenen Platz eines Smartphones optimal zu verwenden. So werden in den weiteren Unterkapiteln 4.3, 4.4 und 4.5 Verfahren vorgestellt, welche den Kontext von Abweichungen in intermodalen Routen erkennen sollen.

4.1 KONTEXT VON DEVIATIONEN IM INTNAV

In der Navigation werden jegliche Abweichungen von der geplanten Route als *Deviationen* bezeichnet, indem ganz klassisch eine Divergenz zwischen der tatsächlichen Position und der erwarteten Position aufgedeckt wird. Wenn beispielsweise ein Autofahrer eine geplante Strecke fährt und von dieser, absichtlich oder unabsichtlich, abweicht, dann bemerkt das Navigationssystem diese Abweichung und meldet sie, zusammen mit einem Alternativvorschlag, dem Autofahrer.

Bei intermodalen Routen gibt es ebenfalls viele verschiedene Situationen in denen Abweichungen von der Route auftreten können. Diese werden nachfolgend anhand eines Beispielszenarios erläutert. Dies dient dazu, im zweiten Schritt die Arten von Abweichungen zu charakterisieren um im nächsten Schritt Methoden zu präsentieren, die während dieser Forschungsarbeit entwickelt wurden, um den Kontext von Abweichungen identifizieren zu können.

Nachfolgend ist eine Modellierung einer intermodalen Route dargestellt (**Abbildung 30**). Dieses Modell stellt eine sehr starke Vereinfachung einer intermodalen Route dar. Die zwei ausgefüllten Kreise, am Anfang und am Ende der Route, markieren dabei den Start- und Endpunkt der Route. Die Route wurde in sieben Segmente (A-G) unterteilt. Diese Segmente bilden die einzelnen Situationen anhand welcher nachfolgend die Abweichungen diskutiert werden. Die Situationen innerhalb der Segmente beschreiben, welche Arten von Abweichungen vorliegen können. Dabei werden sämtliche Möglichkeiten von Abweichungen dargestellt, sodass anschließend die entwickelten Algorithmen für verschiedene Situationen besser nachvollzogen werden können.

Zunächst findet also eine reine theoretische Betrachtung der Situationen samt der Beschreibung des wünschenswerten Verhaltens eines intNAVs wieder. Anschließend wird der entwickelte Algorithmus präsentiert.

Vorbedingung:

Das intNAV kennt die beabsichtigte Route des Benutzers (vgl. **Abbildung 6**). Wichtig ist, dass der Benutzer den Verlauf der Route kennt und das intNAV davon ausgeht, dass der Benutzer diese Route demnächst antreten wird. Die in Kapitel 3 präsentierte TME ist in der Lage die aktuelle Fortbewegungsart des Benutzers automatisiert zu erkennen. Wie beschrieben, ermittelt die TME mittels GPS und den KP die Fortbewegungsarten, zu *Fuß*, *ÖPNV* und *Auto*. Das intNAV lauscht in regelmäßigen Abständen zum einen auf die TME als auch auf die geplante Route. Sobald in einem Zeitpunkt t_i die Transportarten abweichen, soll das intNAV diese feststellen.

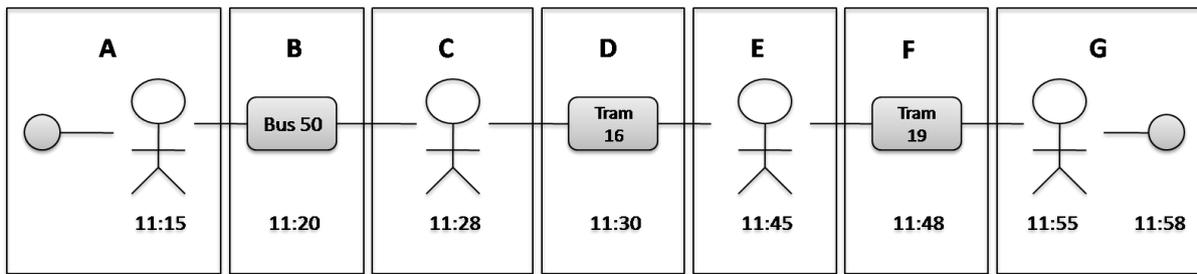


Abbildung 30: Darstellung möglicher Situationen während der Verwendung einer intermodalen Route

Daraufhin wird anhand der Ursache und der Auswirkung der Abweichung entschieden, *wie* und *was* dem Benutzer dabei mitgeteilt werden soll.

Segment A

Der Benutzer soll laut intNAV um 11:15 losgehen, damit er um 11:20 den Bus Nr. 50 an der nächsten Haltestelle erreichen kann. Das intNAV stellt keine Störungen auf dem weiteren Routenverlauf, insbesondere also keine Verspätung des Busses Nr. 50, fest. Dieser Fall, also die Verspätung des Busses, wird später erläutert. Nun sind folgende Situationen denkbar:

A.1 Benutzer macht sich *pünktlich* um 11.15 auf den Weg zur Bushaltestelle *und* wählt dafür auch den möglichst direkten Weg.

A.2 Benutzer macht sich *nicht pünktlich* um 11.15 auf den Weg.

A.3 Benutzer macht sich zwar *pünktlich* um 11.15 auf den Weg, *weicht* aber von der beabsichtigten Route zur Bushaltestelle *ab*, sodass er Gefahr läuft den Bus um 11.20 an der Haltestelle zu verpassen.

Nachfolgend sollen relevante *Information* für die jeweilige Situation diskutiert werden, welche der Benutzer im intNAV angezeigt haben möchte.

Situation A.1 entspricht der vorgegebenen Route auf dem intNAV. Somit wird sich der Benutzer im aktuellen Schritt höchstens die Information darüber wünschen, ob der weitere Verlauf der Route noch aktuell ist oder ob es Probleme, beispielsweise Verspätungen, gibt. Ein Hinweis darüber, dass alles nach Plan verläuft, kann eingeblendet werden. In **Situation A.2** macht sich der Benutzer nicht pünktlich auf den Weg. Da das intNAV von einer Rationalität des Benutzers ausgeht, wird dies als Abweichung angenommen. Wenn das intNAV keine Aktion des Benutzers feststellen konnte, wird eine Neuberechnung der anvisierten Route, ausgehend vom aktuellen Zeitpunkt und Ort des Benutzers, gestartet und dem Benutzer zur Auswahl präsentiert. In **Situation A.3** müsste das intNAV die Abweichung von der Fußgängeroute zur Bushaltestelle erkennen. Nun muss das intNAV den Kontext der Abweichung herauskriegen, also ist sich der Benutzer dieser Abweichung bewusst und wurde diese absichtlich herbeigeführt oder ist die Abweichung unbeabsichtigt entstanden, z.B. weil sich der Benutzer verirrt hat. Je nach Ursache der Abweichung wird diese zusammen mit Alternativen dem Benutzer präsentiert.

Segment B (D und F)

Das Segment B steht stellvertretend für die Segmente D und F, da dort die gleichen Situationen auftreten können. Im Segment B ist der Bus Nr. 50 um 11.20 abgebildet. Damit ist gemeint, dass die intermodale Route für die gewählte Strecke, den Bus Nr. 50 um 11.20 an einer bestimmten Haltestelle vorschlägt. Das intNAV stellt keine Störungen im weiteren Routenverlauf fest. Folgende Situationen sind nun denkbar:

B.1 Der Bus Nr. 50 ist *pünktlich*

B.2 Der Bus Nr. 50 ist *unpünktlich*

B.3 Der Bus Nr. 50 ist *pünktlich*, es kommt jedoch zu *Abweichungen* im Fahrplan.

Analog zur Situation A.1, erwartet auch hier (**Situation B.1**) der Benutzer keine weitere Information, außer der Aktualität im weiteren Routenverlauf. Ein Hinweis darüber, dass alles nach Plan verläuft, kann eingeblendet werden. In der **Situation B.2** hingegen ist nun die Frage, wann das intNAV diese Störung festgestellt hat. Wenn dies um 11.20 der Fall war, dann wird das intNAV den Benutzer unmittelbar darüber in Kenntnis setzen und ausgehend von seinem aktuellen Standort (Bushaltestelle Nr. 50) Alternativen vorschlagen. Wenn das intNAV diese Störung jedoch bereits zu einem früheren Zeitpunkt, festgestellt hat, also während der Benutzer sich bereits zu Fuß zur Bushaltestelle auf den Weg gemacht hat, dann hätte das intNAV die Aktualität der übrigen Transportmittel auf der Route überprüfen müssen. Wenn da keine Verspätungen feststellbar sind, wird ein Schwellenwert ermittelt, bis wann sich der Bus Nr. 50 maximal verspäten kann damit die Route dennoch haltbar ist. Diese maximale Verspätung wird abhängig von alternativen Routen, welche sich in der Nähe des Standortes des Benutzers befinden, gemacht. Dies bedeutet, dass beispielsweise das intNAV festgestellt hat, dass sich der Bus maximal um zwei Minuten verspäten darf (also höchstens um 11:22 Uhr kommen darf), da sonst die Tram Nr. 16 nicht mehr erwischt werden kann. Gleichzeitig jedoch bekommt das intNAV mit, dass eine Alternativroute besteht, um 11:21 mit dem Bus Nr. 177 abzufahren um noch pünktlich zu sein. Die anderen Alternativen führen dazu, dass der Benutzer sein Ziel nicht um 11:58 sondern frühestens um 12:04 erreicht. Wenn sich nun der Bus länger als 2 Minuten verspätet, dann hat der Benutzer die Alternative ebenfalls verpasst und kommt nun mit Sicherheit mindestens um 12:04 an. **Situation B.3** beschreibt den Fall, dass der Bus zwar pünktlich um 11.20 losgefahren ist, jedoch aufgrund von Behinderungen während der Fahrt den Fahrplan nicht pünktlich einhalten kann. Somit ist nicht sichergestellt, dass der Bus um 11:28 Uhr an der, für den Benutzer relevanten, Haltestelle sein wird. Das intNAV soll diese Verspätung während der Fahrt nun erkennen und die Folgen davon ableiten. Entweder die Verspätung des Busses hat Einfluss auf den weiteren Routenverlauf oder nicht. Wenn sie keinen Einfluss hat, dann soll der Benutzer über die aktuelle Situation informiert werden. Es folgt jedoch keine Aktion für den Benutzer. Wenn jedoch die Verspätung Einfluss auf den weiteren Routenverlauf hat, dann wird das intNAV mögliche Alternativen berechnen und dem Benutzer diese mitteilen. In diesem Fall, wird eine Aktion durch den Benutzer nötig sein.

In allen Fällen jedoch stellt sich die Frage nicht danach, ob der Benutzer die Abweichung absichtlich oder unabsichtlich herbeigeführt hat, da der Benutzer für die Deviation in Segment B, D und F nicht verantwortlich ist.

Segment C (und E)

Die Situationen und Abweichungen im Segment C entsprechen denen aus Segment E. Somit gelten diese Situationen und Abweichungen auch für Segment E. Beim dritten Segment soll sich der Benutzer um 11:28 Uhr an einem Umsteigeplatz befinden um von dem Bus Nr. 50 in die Tram-Linie Nr. 16 zu gelangen. Dafür hat er zwei Minuten Zeit. Unter der Annahme, dass im weiteren Routenverlauf keine Verzögerungen entstehen, können folgende Situationen auftreten:

C.1 Der Benutzer kommt *pünktlich* an der Haltestelle an und *erwischt* die richtige Tram Nr. 16.

C.2 Der Benutzer kommt *nicht pünktlich* an der Haltestelle an.

C.3 Der Benutzer kommt *pünktlich* an der Haltestelle an, *läuft jedoch Gefahr* Tram Nr. 16 zu verpassen.

Alle drei Situationen **C.1** - **C.3** sind mit den Situationen von Segment A vergleichbar. Die Folgen, welche das intNAV berechnet sind ähnlich. Die Ursachen für Situationen C.1-C.3 sind jedoch unterschiedlich als bei Segment A. Während im Segment A der Benutzer für alle Situationen selbst verantwortlich ist, ist es im Segment C das Transportmittel, mit welchem der Benutzer zuvor unterwegs war. Dies bedeutet insbesondere, dass in der **Situation C.2** das intNAV die mögliche Verspätung für Segment C im Vorfeld hätte erkennen müssen um dem Benutzer Alternativen vorzuschlagen. Diese Alternativen können nun sein, entweder nichts zu tun, da die Verspätung von dem Transportmittel keine Auswirkung auf den weiteren Verlauf der Route hat oder Berechnung einer alternative Route. In **Situation C.3** stellt das intNAV beispielsweise fest, dass die Person zu lange an einer Stelle verweilt ist und sich nicht zur Tram Nr. 16 bewegt. Der Grund hierfür könnte sein, dass die Person einen Bekannten getroffen hat mit dem sie sich unterhalten hat. Das intNAV berechnet einen Schwellwert ab wann es anfängt, den Benutzer darüber zu informieren, dass sich dieser nun auf den Weg machen soll (gehen oder laufen), da sonst die Tram Nr. 16 verpasst wird. Dabei gibt es nun folgende drei Möglichkeiten. Wenn die Tram pünktlich ist und der Rest der Route keine Probleme aufweist, gibt es keinen Grund den Benutzer über eine Deviation zu informieren. Wenn die Tram jedoch unpünktlich ist, muss das intNAV überprüfen, ob dies eine Auswirkung auf den weiteren Routenverlauf hat. So kann es sein, dass eine 2-Minütige Verspätung der Tram Nr. 16 keine Auswirkung auf den weiteren Routenverlauf hat, da die Tram Nr. 19 immer noch erreicht werden kann. Wie in **Abbildung 30** zu sehen ist, hat die Person drei Minuten Puffer um zur Tram Nr. 19 zu gelangen. Wenn jedoch diese Unpünktlichkeit der Tram Nr. 16 eine Auswirkung auf den weiteren Routenverlauf hat, also dass Tram Nr. 19 nicht mehr erreicht werden kann, so muss das intNAV eine Alternative berechnen und diese dem Benutzer anzeigen. Abhängig von den Auswirkungen fällt die Benachrichtigungsart und –stärke des intNAVs aus.

Segment G

Im letzten Segment hat der Benutzer drei Minuten Zeit um zu seinem Ziel zu gelangen. Das intNAV stellt keine Störungen im weiteren Routenverlauf fest. Dabei können folgende Situationen eintreten:

G.1 Der Benutzer kommt *pünktlich* zu Beginn des Segmentes G an.

G.2 Der Benutzer kommt *nicht pünktlich* zu Beginn des Segmentes G an.

G.3 Der Benutzer kommt zwar *pünktlich* zu Beginn des Segmentes G an, *weicht* jedoch von dem Routenverlauf *ab*.

Im Gegensatz zum Segment A, liegt die Verantwortung für das Eintreten der **Situationen G.1** und **G.2** nicht in der Hand des Benutzers sondern des Transportmittels in dem Segment F. Wenn dieses sich verspätet, dann startet auch die Person im Segment G zu spät. Wenn die Person verspätet in Segment G gelandet ist (**Situation G.2**), dann bedeutet es, dass das intNAV keine bessere Alternative für das Erreichen des Ziels vorgeschlagen hat. Es kann jedoch auch sein, dass die Verspätung sehr gering war, sodass die Person diese durch laufen wieder einholen könnte. **Situation G.3** kann *sowohl* durch Transportmittel aus Segment F als auch durch den Benutzer selbst verursacht sein. Die Ursachen dafür, dass die Person trotz rechtzeitigem Eintreffen beim Segment G verspätet beim Ziel ankommt, sind vielfältig. Sie kann z.B. einen Bekannten getroffen haben, mit dem sie sich unterhält oder beabsichtigt einen anderen POI (z.B. Supermarkt) anlaufen. Abhängig von diesem Kontext soll der Inhalt als auch die Art der Benachrichtigung über eine Abweichung an den Benutzer kommuniziert werden.

Störungen im weiteren Routenverlauf

Über alle bisher beschriebenen Segmente hinweg wurde immer folgender Nebensatz eingefügt: „Das intNAV stellt keine Störungen im weiteren Routenverlauf fest.“ Dies wurde aus dem Grund gemacht, weil sonst keine einfache Segmentierung des Schaubildes (**Abbildung 30**) möglich wäre. Zusätzlich gäbe es eine große Anzahl an möglichen Situationen, welche hier beschrieben werden müssten, die jedoch den Aufwand mit dem erzielten Ergebnis dabei nicht rechtfertigen würden. Dies bedeutet jedoch nicht, dass das hier entwickelte intNAV nicht in der Lage wäre, Routenstörungen während der Fahrt festzustellen. Ein solches intNAV würde die unter [2.1.2](#) genannten Anforderung 3 widersprechen und wäre nicht zielführend.

Nachfolgende **Abbildung 31** stellt nun einen Auszug aus **Abbildung 30** dar, wobei diesmal Routenstörungen auf dem weiteren Routenverlauf möglich sind. Nachfolgend werden die Situationen und die Informationen, welche der Benutzer bei einem intNAV sehen sollte, beschrieben.

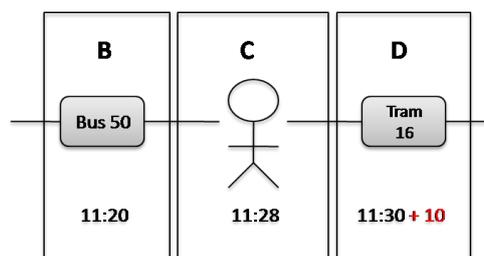


Abbildung 31: Auszug einer intermodalen Route

Während der Benutzer um 11:24 mit dem Bus Nr. 50 die beabsichtigte Route (Segment B) fährt, stellt das intNAV eine Störung des weiteren Routenverlaufs fest, weil Tram Nr. 16 eine Verspätung von 10 Minuten meldet. Das intNAV soll nun zunächst abgleichen, ob eine Verspätung der Tram Nr. 16 um 10 Minuten eine Auswirkung auf den weiteren Routenverlauf hat. Wenn dies *nicht* der Fall wäre, weil beispielsweise die Anschluss-Tram Nr. 19 ebenfalls eine Verspätung verzeichnet oder laut Fahrplan erst in 15 Minuten abfährt, dann wird das intNAV

den Benutzer über die jeweiligen Abweichungen auf der Route in Kenntnis setzen aber keine **Aktion** fordern sondern dem Benutzer empfehlen, die Route beizubehalten. Wenn jedoch die anderen Transportmittel auf dem Rest der Route pünktlich sind, dann hat die Verspätung Auswirkungen auf den weiteren Routenverlauf und somit kann die Tram Nr. 19 nicht mehr rechtzeitig erreicht werden. Das intNAV berechnet nun mögliche Alternativen vom *aktuellen Ort* des Benutzers. Das intNAV hat mittels TME den Bus Nr. 50 und die nächste Haltestelle identifiziert, bei welcher der Benutzer aussteigen könnte. Basierend auf dieser Information (nämlich der nächsten Haltestelle des aktuell gefahrenen Transportmittels), berechnet das intNAV eine Routenalternative und stellt beispielsweise fest, dass der Benutzer an der nächsten Haltestelle aussteigen kann und dann z.B. einen Bus Nr. 177 nehmen kann um dann weiter ohne Umsteigen sein Ziel um 11:59 zu erreichen.

Eine ähnliche Logik ist anwendbar, wenn sich in **Abbildung 31** der Bus, in welchem sich die Person aktuell befindet, nicht mehr an den Fahrplan hält, beispielsweise verursacht durch einen Stau und die Person dadurch Gefahr läuft die Tram Nr. 16 zu verpassen.

4.2 ARTEN VON DEVIATIONEN IM INTNAV

Nachfolgend sind alle angesprochenen Situationen tabellarisch auf Grund der besseren Übersicht dargestellt. Die Situationen wurden dabei in zwei Tabellen unterteilt. Einmal in **Tabelle 9**, welche sämtliche Situationen anzeigt, die **vor** einem Routenabschnitt beginnen können und in **Tabelle 10**, welche sämtliche Situationen anzeigt, die **während** eines Routenabschnittes stattfinden können. Ein Routenabschnitt ist immer durch einen POI von dem nächsten Routenabschnitt getrennt, also beispielsweise durch eine Haltestelle. Die Tabellen **Tabelle 9** und **Tabelle 10** dienen der Übersichtlichkeit der im Vorfeld ausführlich beschriebenen Segmente und deren Situationen. Dabei wurden aus Gründen dieser Übersichtlichkeit sämtliche Sonderfälle herausgenommen. Diese finden sich jedoch, in der ausführlichen Beschreibung der Situationen im vorhergehenden Abschnitt wieder.

Ein Routenabschnitt bezeichnet den Routenverlauf zwischen zwei POIs. Segmente A, C, E und G sind beispielsweise Routenabschnitte aus Sicht des Benutzers. Segmente B, D und F bezeichnen Routenverläufe von den Transportmitteln. Folgende **Tabelle 9** zeigt nun alle Situationen auf, die vor Beginn eines Routenabschnittes in einem intNAV auftreten können.

Verhalten von		Auswirkung auf intNAV
Benutzer	nächstes Transportmittel	
pünktlich	pünktlich	Routenverlauf weiter verfolgen.
pünktlich	nicht pünktlich	Schwellwert errechnen, bis wann diese Verspätung in Hinblick auf den aktuellen Routenverlauf hinnehmbar ist. Wenn Schwellwert überschritten wurde, Alternative berechnen.
nicht pünktlich	pünktlich	Kontext von Abweichung verstehen. Schwellwert errechnen, bis wann diese Verspätung in Hinblick auf den aktuellen Routenverlauf hinnehmbar ist und Benutzer ggf. auffordern

		schneller zu gehen. Wenn Schwellwert überschritten wurde, Alternative berechnen.
nicht pünktlich	nicht pünktlich	Bei einer beabsichtigten Verspätung durch den Benutzer weist das intNAV über die Verspätung des nachfolgenden Transportmittels weniger penetrant hin als bei einer unbeabsichtigten Verspätung des Benutzers, während es gleichzeitig den Schwellwert errechnet, bis wann diese Verspätung in Hinblick auf den aktuellen Routenverlauf hinnehmbar ist. Wenn dieser Schwellwert überschritten wurde, werden Alternativen berechnet.

Tabelle 9: Mögliche Situationen vor einem Routenabschnitt

Ein Benutzer kann von dem vorgegebenen Routenverlauf abweichen, indem dieser beispielsweise einen Schleichweg verwendet, einen Bekannten trifft und sich länger mit ihm unterhält oder absichtlich einen anderen POI ansteuert (beispielsweise Supermarkt). Ein Transportmittel kann von seinem Routenverlauf abweichen, wenn es beispielsweise auf Grund von Störungen oder Staus den Fahrplan nicht einhalten kann. Dabei ist wichtig zu erwähnen, dass die Situationen, welche in der nachfolgenden **Tabelle 10** beschrieben sind, sich nicht auf den unmittelbar anstehenden Routenabschnitt beziehen müssen sondern auf alle Routenabschnitte r_i der Route R .

Verhalten von Benutzer/Transportmittel	Auswirkung auf intNAV
Hält sich an den vorgegebenen Routenverlauf	Routenverlauf weiter verfolgen.
Weicht vom vorgegebenen Routenverlauf ab	Kontext von Abweichungen des Benutzers verstehen. Schwellwert errechnen, bis wann diese Verspätung in Hinblick auf den aktuellen Routenverlauf hinnehmbar ist. Wenn Schwellwert überschritten wurde, Alternative berechnen.

Tabelle 10: Mögliche Situationen während eines Routenabschnittes

Bei allen Segmenten wurden nun mögliche Situationen aufgezeigt, welche bei einem intNAV auftreten können und welche Arten von Deviationen es somit in intNAVs gibt. Dabei fällt auf, dass es zwei sogenannte Fehlerquellen gibt, welche in den präsentierten Situationen Abweichungen verursachen können, nämlich den Benutzer selbst und die entsprechenden Transportmittel auf der Route. Diese werden nachfolgend, in Kapitel 4.2.1 und 4.2.2 näher betrachtet.

4.2.1 Aktive und passive Deviationen

Bei einem intNAV wird immer davon ausgegangen, dass ein Benutzer dieses Navigationssystem aktiv betätigt hat. Aus diesem Grund werden Abweichungen, welche aufgrund des Verhaltens des Benutzers ausgelöst wurden, als **aktive Abweichungen** bezeichnet. Im Gegensatz dazu stehen **passive Abweichungen** welche durch das Verhalten der Umwelt (also eines Transportmittels) zu einer Deviation im Routenverlauf führen. Diese Kategorisierung kann nun weiter unterteilt werden. Beispiele für aktive und passive Abweichungen wurden im vorhergehenden Kapitel 4.1 beschrieben. Aktive und passive Abweichungen führen nicht immer dazu, dass das intNAV eine Neuberechnung der Route vornimmt. Nur bei passiven Abweichungen, bei denen keine Möglichkeit mehr besteht, die aktuelle Route weiter zu verwenden, löst das intNAV eine Neuberechnung aus und präsentiert dem Benutzer dabei die Alternativen. Bei aktiven Abweichungen geht das intNAV davon aus, dass der Benutzer sich über die getätigte Abweichung im Klaren ist und präsentiert weniger penetrant die Alternativen, dennoch findet aufgrund der aktiven Abweichung eine Neuberechnung statt.

Aktive Abweichung

Aktive Abweichungen sind Abweichungen, welche durch den Benutzer selbst (bewusst oder unbewusst) initiiert werden.

Passive Abweichung

Passive Abweichungen sind Abweichungen, welche nicht durch den Benutzer selbst sondern durch die Umwelt (die Transportmittel) initiiert werden.

4.2.2 Echte und unechte Deviationen

Es gibt jedoch Abweichungen, egal ob passiv oder aktiv, welche trotzdem das Ausführen der ursprünglichen Route nicht behindern. Einige Beispiele wurden im vorhergehenden Kapitel bereits genannt. Ein Beispiel für eine aktive Abweichung, welche jedoch keine Auswirkung für den weiteren Verlauf der Route hat, ist wenn die Person einen Schleichweg kennt um zum vorgeschlagenen POI zu gelangen. Eine Verspätung des Verkehrsmittels, welches jedoch ebenfalls keine Auswirkung auf den weiteren Verlauf der aktuellen Route hat, ist ein Beispiel für eine passive Abweichung welche keine Neuberechnung der Route mit sich zieht. Solche Abweichungen wurden im Rahmen der vorliegenden Arbeit als **unechte Abweichungen** genannt; **echte Abweichungen** sind dagegen diejenigen, welche eine Neuberechnung der Route mit sich führen.

Echte Abweichung

Eine echte Abweichung ist eine erkannte (aktive oder passive) Abweichung, welche unmittelbaren Einfluss auf den weiteren Routenverlauf hat.

zu einer Neuberechnung der Route führt.

eine Aktion vom Benutzer verlangt.

Unechte Abweichung

Eine unechte Abweichung ist eine erkannte (aktive oder passive) Abweichung, welche keinen unmittelbaren Einfluss auf den weiteren Routenverlauf hat.

nicht zu einer Neuberechnung der Route führt.

keine Aktion vom Benutzer verlangt.

Unechte Abweichungen können jedoch zu echten Abweichungen werden, wenn die Abweichung so lange bestehen bleibt, dass sie einen unmittelbaren Einfluss auf den weiteren Routenverlauf hat.

In **Abbildung 32** findet sich eine Unterteilung der jeweiligen Arten, welche im Rahmen dieser Arbeit ausgearbeitet wurden. Die Unterteilung der Deviationsarten beeinflusst den *Inhalt* (was) sowie die *Art* (wie), wie eine Deviation dem Benutzer mitgeteilt werden soll. So soll eine Abweichung, welche vom intNAV erkannt wurde und welche dazu führt, dass der Benutzer so schnell wie möglich an der nächsten Haltestelle den Bus verlässt um in die nächste Tram zu wechseln, anders kommuniziert werden als eine Abweichung, welche lediglich dazu führt, dass der Benutzer eine Minute länger an der Haltestelle warten muss aber der weitere Routenverlauf sich nicht ändert und sein Ziel jedoch trotzdem pünktlich erreicht werden kann.

Die Arten wie diese Abweichungen dem Benutzer am besten kommuniziert werden, werden in Kapitel 6 diskutiert. Im Rahmen des vorliegenden Kapitels werden Algorithmen präsentiert, wie absichtliche von unabsichtlichen aktiven Abweichungen erkannt werden können. In **Abbildung 32** sind passive Abweichungen ausgegraut. Dies liegt daran, dass es keine Herausforderung ist, passive Abweichungen zu erkennen, da das vorliegende entwickelte intNAV Zugriff zu Echtzeitinformationen der Transportmittel hat. Auch die Berechnung, ob es eine echte oder eine unechte passive Abweichung ist, ist ebenfalls trivial. Passive Abweichungen, wie weiter vorne erwähnt, werden durch die Umwelt, also in diesem Falle, die Transportmittel initiiert. Ursache hierfür können für den ÖPNV Verspätungen, Störungen oder Ausfälle sein. Die Information für diese Abweichungen wird extern bezogen, dies bedeutet wenn ein ÖPNV-Verkehrsmittel sich verspätet, wird dies mit Hilfe einer Schnittstelle erkannt. In weiterführenden Arbeiten, kann sich der Vorhersage einer Erkennung von Verspätungen in ÖPNV-Netzen gewidmet werden und zusammen mit dem hier entwickelten Algorithmus zur Vorhersage des Benutzerverhaltens, zu einem besseren Vorhersagemodell ausgeführt werden.

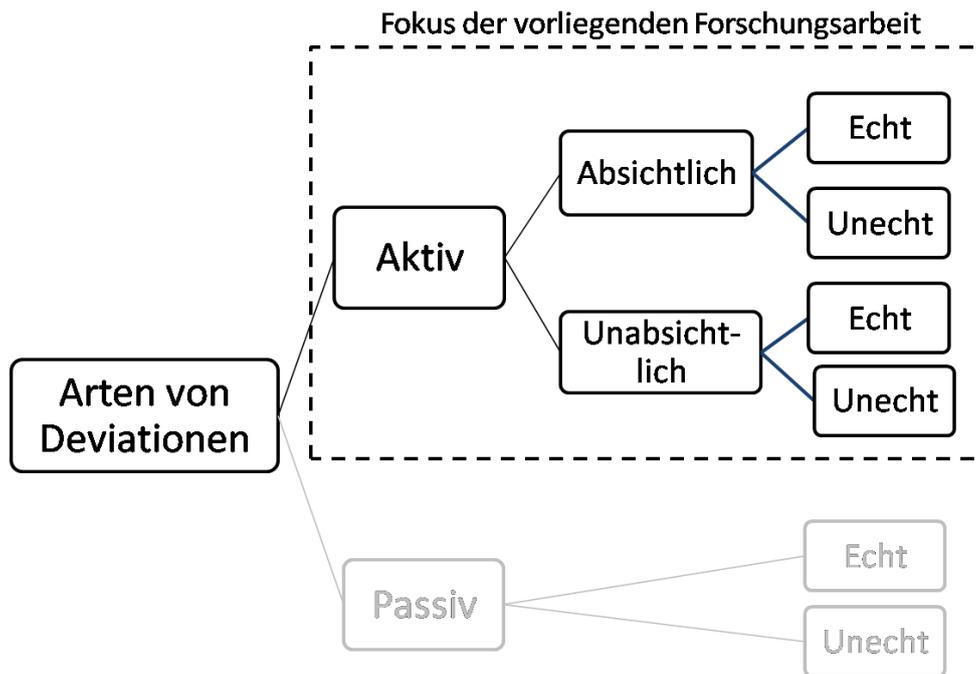


Abbildung 32: Arten von Deviationen

Es kann sein, dass der Benutzer durch Ausnutzen von Schleichwegen vielleicht sogar schneller zur Bushaltestelle gelangt. In diesem Fall, wäre es kontraproduktiv den Benutzer dazu zu zwingen, doch den berechneten Weg zu nehmen, da eine aktive Abweichung festgestellt wurde.

Um die Frage nach dem *wie* die Information über eine Deviation dem Benutzer auf seinem Smartphone mitgeteilt werden soll, widmet sich Kapitel 6. Dort werden zwei Studien beschrieben, welche den intNAV Prototypen getestet haben und insbesondere, neben der Funktionalität, den Zeitpunkt und die Unterbrechungsart bewertet wurde. Auch hängt der Inhalt der Information, welche das intNAV bei einer erkannten Abweichung dem Benutzer mitteilt, von der Art der Abweichung ab. So wird eine echte Abweichung immer mindestens einen Alternativvorschlag für die Route enthalten, welche eine Aktion vom Benutzer fordert, wohingegen eine unechte Abweichung dies nicht, zumindest nicht sofort, enthält.

Somit kann zusammengefasst werden, dass die Ursachen für Deviationen zum einen beim Benutzer und zum anderen bei den verschiedenen Transportmitteln liegen. Zusätzlich gibt es echte Abweichungen welche immer eine Aktion vom Benutzer fordern, wohingegen unechte Abweichungen dies nicht, zumindest nicht sofort, tun. Aus einer unechten Abweichung kann aber eine echte Abweichung entstehen, wenn die Art der Abweichung (aktiv oder passiv) weiterhin bestehen bleibt. Die nachfolgenden Unterkapitel zeigen nun den entwickelten Ansatz, welcher Abweichungen vom Benutzer möglichst früh prognostizieren kann.

4.3 HYPOTHESEN

[Dieses Unterkapitel basiert auf der **Veröffentlichung** von „*Intelligent Deviation Perception and Prediction of Deviaton Probability*“, Olga Birth, Yu Fei, publiziert in *International Conference on Information Processing and Control Engineering*, Moskau, April 2015]

Die im vorangegangenen Kapitel angesprochenen Arten von Abweichungen sollen mittels eines intNAVs rechtzeitig erkannt und im Falle einer echten Abweichung dem Benutzer Alternativen aufgezeigt werden. Die Abweichungen auf einem intNAV sind bedingt durch eine Vielzahl an unterschiedlichen Umweltfaktoren, welche sich im Bruchteil einer Sekunde verändern können. Wie bereits beschrieben handelt es sich bei Umweltfaktoren um das Verhalten sämtlicher an einem Ort vorhandenen Transportmittel und deren Verbindungen untereinander. Die Anzahl der Alternativen, der sogenannte Ergebnisraum, steigen überproportional zu der Anzahl der verfügbaren Transportmittel. Bei einem Autonavigationssystem ist dieser Ergebnisraum viel geringer und bezieht sich nur auf Kombinationen der Autorouten. Bei einem Autonavigationssystem beziehen sich die Umweltfaktoren auf sämtliche Störungen innerhalb der Auto-Route verursacht durch Staus, Baustellen, Umleitungen, Unfälle etc. und auf das abweichende Verhalten des Benutzers. Bei einem intNAV beziehen sich die Umweltfaktoren ebenfalls auf sämtliche Störungen auf der aktuellen intermodalen Route, Verspätungen, Ausfälle und das Benutzerverhalten, jedoch bezieht der Ergebnisraum **zusätzlich** sämtliche an einem Ort zur Verfügung stehenden Transportmittel mit ein. In einer Stadt wie München, können sich die daraus resultierenden Alternativrouten innerhalb von Sekunden ändern, da mit der Anzahl der Transportmittel, die Anzahl der Störungen zunehmen kann. Somit sind die Entscheidungen eines intNAVs sehr volatil, da es zum einen von sehr vielen Umweltfaktoren beeinflusst wird und zum anderen die Zusammenhänge zwischen den einzelnen Umweltfaktoren in Betracht ziehen muss.

Um ein ständiges hin und her („jitter“) der Empfehlungen im intNAV zu vermeiden, sollen mögliche Abweichungen nicht erst in dem Moment erkannt werden, wenn diese eintreten, sondern möglichst früh vorhergesagt werden. Im Rahmen der vorliegenden Arbeit wurde ein Ansatz entwickelt, welcher Abweichungen in intermodalen Routen vorhersagen kann, sodass diese nicht erst in dem Moment entdeckt und behoben werden, wenn diese gerade eingetreten sind und im nächsten Moment bereits nicht mehr aktuell sind. Dabei ist zunächst wichtig zu verstehen, dass es, wie im vorangegangenen Kapitel beschrieben wurde, zwei Ursachen für eine Abweichung gibt, einmal den Benutzer und einmal die Umwelt, also die Transportmittel. Der hier entwickelte Ansatz fokussiert sich auf die Vorhersage einer Abweichung des Benutzers von der festgelegten Route, da die Forschung in diesem Bereich noch Lücken aufweist, wohingegen zur Prognose von Abweichungen von Verkehrsmittel bereits viele Ansätze in der Literatur existieren. Nachfolgende Hypothesen liefern Rahmenbedingungen und Annahmen, welche für den hier entwickelten Algorithmus gelten um Abweichungen des Benutzers vorhersagen zu können.

Hypothese I – Rationalität

Der Benutzer handelt rational. Der Benutzer wählt eine Route im intNAV aus, welches aus seiner Sicht die beste Kombination aus seinem Start- und Zielort, der Dauer und der Transportmittel darstellt. Der Benutzer hält sich an die vorgeschlagene Route und bewegt sich dementsprechend. Obwohl es vorkommen kann, dass der Benutzer

seine Abweichung nicht sofort bemerkt, wenn er auf diese Abweichung aufmerksam gemacht wird, soll der Benutzer diese korrigieren.

Hypothese II – Begrenzte Ziele

Die Ziele welche der Benutzer ansteuert sind begrenzt. Basierend auf Hypothese I befindet sich das Ziel des Benutzers in der Nähe der gewählten Zielposition. Der entwickelte Algorithmus soll eine Toleranz bezüglich der gesetzten Zielposition besitzen, sodass beispielsweise Parkplätze in der Umgebung oder Treffen von Bekannten nicht gleich als Deviation vom Algorithmus erkannt wird, sondern diese herausfiltert.

Hypothese III – Regularität

Der entwickelte Algorithmus geht von normalen Bewegungsabläufen aus. Die Geschwindigkeit (im Fußgängermodus), die maximale unscharfe („fuzzy“) Abweichungsdistanz und –zeit werden als stabile Werte behandelt, so dass minimale Fluktuationen und Oszillationen statistischen Gesetzen folgen.

Mit Hilfe dieser Hypothesen wird die nachfolgende Methodik zur Prognose von Abweichungen in intNAVs beschrieben.

4.4 ERKENNUNG DES KONTEXTES VON DEVIATIONEN

Eine Deviation kann als ein Konstrukt angesehen werden, welches aus drei Dimensionen besteht: Abweichung in der Zeit, im Raum und in der Logik, wobei die Logik-Dimension in der Literatur bis jetzt einzigartig ist. Im intNAV kann die Erkennung einer Deviation, ähnlich wie die TME (vgl. Kapitel 3) als Blackbox angesehen werden, welche die Umwelt und die historischen Benutzerdaten als Input verwendet und die Wahrscheinlichkeit einer Abweichung als Output ausgibt. Folgende **Abbildung 33** zeigt dabei schematisch die Abläufe innerhalb der erwähnten Blackbox.

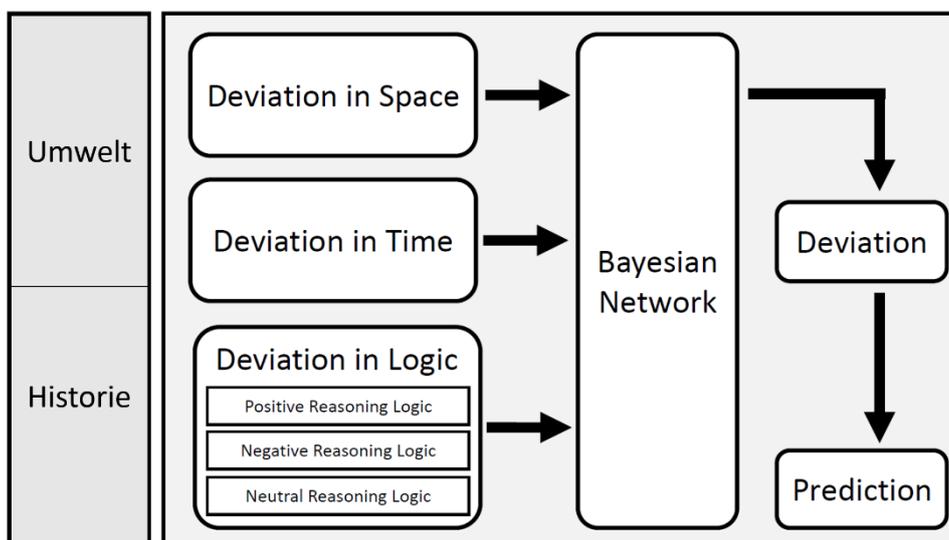


Abbildung 33: Überblick über die Architektur zur Prognose von Anomalien

In dem Architekturbild werden die angesprochenen drei Dimensionen Raum, Zeit und Logik als Input für ein Bayes'sches Netz verwendet um die Abweichung eines Benutzers in intermodalen Routen vorherzusagen. Die

Arbeitsweise innerhalb der jeweiligen Dimensionen sowie innerhalb des Bayes'schen Netzwerks wird in den folgenden vier Unterkapiteln erklärt.

4.4.1 Vorhersage der Wahrscheinlichkeit einer Deviation mit Hilfe eines Bayes'schen Netzwerks

Wie in **Abbildung 33** zu sehen ist, übernimmt das Bayes'sche Netz die gesamthafte Erkennung einer Abweichung. Da der in dieser Arbeit entwickelte Algorithmus die Vorhersage einer möglichen Deviation mit Hilfe der drei Dimensionen, Raum (Kapitel 4.4.2), Zeit (Kapitel 4.4.3) und Logik (Kapitel 4.4.4) bestimmt, müssen die verschiedenen Abweichungen in Verbindung zu einander gebracht werden.

Eine Deviation wird mittels der Variablen Y dargestellt. Eine Abweichung wird somit mittels $Y = 1$ und analog dazu keine Abweichung mittels $Y = 0$ dargestellt. Die Eingangsgrößen für das Bayes'sche Netz sind somit folgende:

Eine Abweichung in der **Raumdimension** wird mit $x = \theta$ definiert. Basierend auf der historischen Verteilung kann die bedingte Wahrscheinlichkeit für eine Abweichung im Raum folgendermaßen berechnet werden:

$$P_{(x \leq \theta | Y=1)}, P_{(x \leq \theta | Y=0)}$$

Eine Abweichung in der **Zeitdimension** wird als $x = \varphi$ definiert. Basierend auf der historischen Verteilung, kann die bedingte Wahrscheinlichkeit für eine Abweichung in der Zeit folgendermaßen berechnet werden:

$$P_{(x \leq \varphi | Y=1)}, P_{(x \leq \varphi | Y=0)}$$

Eine Abweichung in der **Logik** wird als $x = \omega$ definiert. Basierend auf der historischen Verteilung, kann die bedingte Wahrscheinlichkeit für eine Abweichung in der Logik folgendermaßen berechnet werden:

$$P_{(x \leq \omega | Y=1)}, P_{(x \leq \omega | Y=0)}$$

Somit kann die Wahrscheinlichkeit einer möglichen Abweichung folgendermaßen berechnet werden:

$$P(x; \theta, \varphi, \omega | Y = 1) = P_{(x \leq \theta | Y=1)} * P_{(x \leq \varphi | Y=1)} * P_{(x \leq \omega | Y=1)} * P(Y = 1)$$

Analog dazu erfolgt die Berechnung der Wahrscheinlichkeit, wenn keine Deviation erkannt wird:

$$P(x; \theta, \varphi, \omega | Y = 0) = P_{(x \leq \theta | Y=0)} * P_{(x \leq \varphi | Y=0)} * P_{(x \leq \omega | Y=0)} * P(Y = 0)$$

Die Erkennung einer Abweichung kann mittels eines Vergleiches von $P(x; \theta, \varphi, \omega | Y = 1)$ und $P(x; \theta, \varphi, \omega | Y = 0)$ erfolgen.

4.4.2 Deviation in der Raumdimension

Bei einer Deviation im Raum versteht man die Abweichung eines Benutzers von der vorgesehenen Route. Hier kommt **Hypothese I** zum Tragen, da von einer rationalen Entscheidung des Benutzers ausgegangen wird und somit angenommen wird, dass der Benutzer sich normalerweise an die gewählte Route hält. Wie weiter oben in Kapitel 4.2.1 beschrieben steht, bezieht sich die vorliegende Arbeit auf die Vorhersage einer aktiven Abweichung. Die aktuelle Position eines Benutzers wird mittels GPS-Rohdaten in Echtzeit bestimmt. Unabhängig von den

anderen Abweichungen, Zeit und Logik, müssen Regeln aufgestellt und verfolgt werden, um Kriterien zu definieren, wie der Ausmaß einer Abweichung im Raum bestimmt werden kann.

Je weiter die Entfernung zwischen der aktuellen Position des Benutzers und der vorgeschlagenen Route ist, desto höher ist die Wahrscheinlichkeit, dass der Benutzer abgewichen ist.

Die abgewichene Distanz soll in Bezug auf die Gesamtdauer gewichtet werden. Dies bedeutet, dass die Wahrscheinlichkeit einer Abweichung im Raum dynamisch ist.

Jedes Ergebnis der Berechnung der Wahrscheinlichkeiten für eine Abweichung soll für künftige Auswertungen berücksichtigt werden.

Wie in 4.4.1 Vorhersage der Wahrscheinlichkeit einer Deviation mit Hilfe eines Bayes'schen Netzwerks definiert, beschreibt $P_{(x=\theta)}$ die Wahrscheinlichkeit für eine Abweichung in der Raumdimension. Die gesamte Reisedistanz und die wahrscheinliche abgewichene Distanz werden als L und als θ dargestellt. Die **Wahrscheinlichkeit** für die **Abweichung in der Raumdimension** erfolgt somit:

$$P_{(x=\theta)} = -\left(\frac{1}{2}\right)^{\frac{\pi^2\theta}{L}} + 1$$

Um den Zusammenhang der Wahrscheinlichkeit einer Abweichung in der Raumdimension intuitiver darzustellen, wurden in **Abbildung 34** als Inputparameter die gesamte Soll-Reisedistanz in Metern und die wahrscheinliche abgewichenen Distanz in Metern verwendet. Der Abbildung lässt sich entnehmen, dass die Wahrscheinlichkeit für eine Abweichung höher wird wenn die abgewichene Distanz von der Soll-Reiseroute signifikant ist. Um die Eingabe der zwei Inputs zu vereinfachen, wurde hier ein Verhältnis angegeben. Der angegebene Wert 3.0 auf der X-Achse bedeutet in diesem Beispiel $\theta/L = 30\%$.

Das Ergebnis der Berechnung einer möglichen Abweichung wird als eine Menge $\{Y\}$ bezeichnet und $Y=1$ als Abweichung, sowie $\sum n$ als die Anzahl der möglichen Abweichungen $|\{Y\}|$, entsprechend $|\{Y = 0\}|$ oder auch n_0 als die Anzahl der Nicht-Abweichungen und $|\{Y = 1\}|$ oder n_1 als die Anzahl der Abweichungen. Somit folgt:

$$\sum n = |\{Y\}| = |\{Y = 0\}| + |\{Y = 1\}| = n_0 + n_1$$

Gegeben sei eine potentielle Abweichung in der Raumdimension als $P_{(x=\theta_0)}$, so kann $\{Y = 1\}$ weiter aufgesplittet werden in $\{Y = 1\} = \{x|x \in [0, \theta_0], Y = 1\} \cup \{x|x \in (\theta_0, +\infty), Y = 1\}$.

Somit kann die Wahrscheinlichkeit für eine potentielle Abweichung in der Raumdimension von $P_{(x \leq \theta_0 | Y=1)}$ und $P_{(x \leq \theta_0 | Y=0)}$ folgendermaßen dargestellt werden:

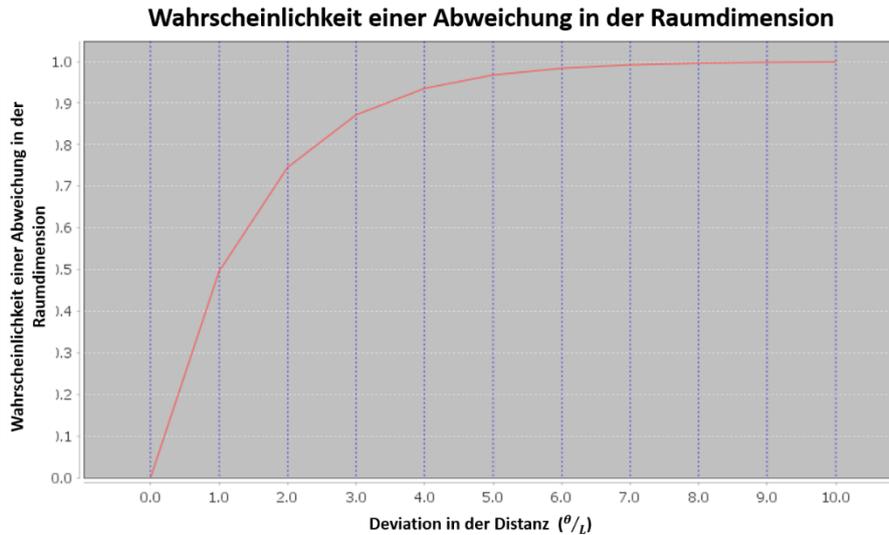


Abbildung 34: Wahrscheinlichkeit einer Abweichung in der Raumdimension

$$P_{(x \leq \theta_0 | Y=1)} = \frac{P(Y = 1 | x \ll \theta_0) * P(x \ll \theta_0)}{P(Y = 1)} = \frac{|\{x | x \in [0, \theta_0], Y = 1\}|}{n_1}$$

$$P_{(x \leq \theta_0 | Y=0)} = \frac{P(Y = 1 | x \ll \theta_0) * P(x \ll \theta_0)}{P(Y = 0)} = \frac{|\{x | x \in [0, \theta_0], Y = 0\}|}{n_0}$$

Eigenschaften der Funktion

Monotonie

Die Ableitung der Funktion $P_{(x=\theta)}$ beträgt

$$\frac{dP(x = \theta)}{dx} = \frac{d\left(-\left(\frac{1}{2}\right)^{\frac{\pi^2 \theta}{L}} + 1\right)}{d\theta} = -\frac{\pi^2}{L} \ln\left(\frac{1}{2}\right) \left(\frac{1}{2}\right)^{\frac{\pi^2 \theta}{L}} \geq 0, \theta \in [0, +\infty),$$

Die Monotonie der Funktion wächst im Bereich von $[0, +\infty)$ und liegt zwischen $[0, 1]$. Da die Funktion eine inverse Funktion im Bereich $[0, 1]$ enthält, besitzt die Funktion und der Bereich eine 1:1-Beziehung.

Konvexität

Unter der Annahme, dass $\frac{\theta}{L}$ eine unabhängige Variable ist, gilt für die Konvexfunktion immer $f\left(\frac{x_1+x_2}{2}\right) > \frac{f(x_1)+f(x_2)}{2}$ im Bereich von $[0, 1]$. Da für die Funktion $P_{(x)} = x, x_1 = 0$ und $P_{(x)} > x, x_2 = 1, P_{(x)} - x \geq 0$ gilt, befindet sie sich immer im Bereich $[0, 1]$.

Erhöhung der Effizienz mittels Gaußscher Verteilung

Um eine hohe Genauigkeit der Wahrscheinlichkeitsberechnung der Deviation zu erhalten, wird eine große Anzahl an historischen Benutzerdaten benötigt. Dies wirkt sich negativ auf die Systemeffizienz und den –speicher aus. Jede Benutzerbewegung wird unabhängig zu den anderen aufgezeichneten Bewegungen angesehen. Jede Bewegung kann als zufälliger Input angesehen werden. Basierend auf **Hypothese III** folgt daraus, dass die

Verteilung der Wahrscheinlichkeit einer Abweichung in der Raumdimension als Gaußsche Verteilung angesehen werden kann.

Für die Gaußsche Verteilung $x \sim N(\mu, \sigma^2)$ mit den Parametern μ und σ^2 gilt:

$$\mu_0 = \frac{\sum x}{n_0}, x \in \{x|Y = 0\},$$

$$\sigma_0^2 = \frac{\sum(x-\bar{x})^2}{n_0}, x \in \{x|Y = 0\},$$

$$\mu_1 = \frac{\sum x}{n_1}, x \in \{x|Y = 1\},$$

$$\sigma_1^2 = \frac{\sum(x-\bar{x})^2}{n_1}, x \in \{x|Y = 1\},$$

Wenn nun eine neue potentielle Abweichung in der Raumdimension $x_t = P_{(x_t=\theta)}$ kommt, kann ein Update an den historischen Daten folgendermaßen vollzogen werden:

$$\mu'_0 = \frac{\sum x'}{n'_0} = \frac{\sum x + x_t}{n'_0} = \frac{\mu_0 n_0 + x_t}{n_0 + 1},$$

$$\sigma'_0{}^2 = \frac{\sum(x-\bar{x}')^2}{n'_0} = \frac{\sum(x-\bar{x})^2}{n_0 + 1},$$

$$\mu'_1 = \frac{\sum x'}{n'_1} = \frac{\sum x + x_t}{n'_1} = \frac{\mu_1 n_1 + x_t}{n_1 + 1},$$

$$\sigma'_1{}^2 = \frac{\sum(x-\bar{x}')^2}{n'_1} = \frac{\sum(x-\bar{x})^2}{n_1 + 1}.$$

Mit Hilfe der Gaußschen Distribution können die historischen Benutzerdaten durch zwei Parameter dargestellt und aktualisiert werden.

Zusätzlich ist das System in der Lage eine mögliche Abweichung frühzeitig zu entdecken. Wenn der Benutzer bereits $\frac{\theta}{L} \gg 0.5$ abgewichen ist, macht es fast keinen Unterschied zu einer Abweichung von 0.7 oder 0.6. Aus diesem Grund reagiert die beschriebene Funktion möglichst frühzeitig auf mögliche Abweichungen. Dies bedeutet, dass $P_{(x=0)}$ möglichst frühzeitig hoch gewichtet werden soll als zu einem späteren Zeitpunkt. Eine konvexe Funktion erfüllt die Bedingungen. Aus den beschriebenen Eigenschaften, Monotonie und Konvexität, wurde eine Funktion erstellt, welche eine unabhängig Variable für die Gaußsche Verteilung darstellt.

4.4.3 Deviation in der Zeitdimension

Unter Berücksichtigung eines bestimmten Zeitintervalls wird der Rest der Reisezeit in Hinblick auf das Ziel geschätzt. Dabei hängt die Deviation in der Zeitdimension von zwei Parametern ab: davon ob sich der Benutzer hinter der geschätzten Reisezeit befindet und der Zeit, wann der Benutzer von der normalen Reiseroute abweicht. In Hinblick auf die korrekte Berechnung der Abweichung in der Zeitdimension werden zunächst folgende Regeln etabliert:

Die Wahrscheinlichkeit für eine Abweichung in der Zeit steigt, wenn die geschätzte Reisezeit welche an einer Echtzeit-Position gemessen wurde einen Schwellwert, welcher dynamisch von der ursprünglich berechneten Reisezeit von der Startposition abhängig ist, überschreitet.

Gleichzeitig setzt ein Timer ein, wenn sich die Distanz zwischen dem Benutzer und der vorgeschlagenen Route über einem dynamisch bestimmten Schwellwert befindet, welcher die potentielle Abweichung in der Zeit misst. Dabei bezieht der dynamische Schwellwert die Ungenauigkeit des GPS in die Berechnung mit ein. Zusätzlich kann dieser Schwellwert damit umgehen, dass der Benutzer auf verschiedenen Seiten des Bürgersteigs gehen kann. Im Gegensatz zu Fahrzeugen, welche Landesabhängig entweder rechts oder links fahren dürfen, kann der Fußgänger sich aussuchen, ob er nun rechts oder links von der Straße geht. Dies kann jedoch einen Fehlerfaktor hervorrufen.

Die gesamte Reisezeit wird mit $\sum T$ beschrieben, welche die Reisezeit über alle Wegsegmente berücksichtigt. Die *geschätzte* Reisezeit wird mittels T_{Est} und die tatsächliche Reisezeit mit T_{Act} beschrieben. Für eine Abweichung in der Zeit, gilt folgendes Kriterium:

$$\frac{|T_{Est} - T_{Act}|}{\sum T} \gg \left(\frac{T_{Act}}{\sum T}\right)^2$$

Dies ist äquivalent zu

$$T_{Est} \gg \frac{T_{Act}^2}{\sum T} + T_{Act}$$

Somit wird die **Wahrscheinlichkeit für eine Abweichung in der Zeitdimension** $P_{(x=\varphi)}$ mittels folgender Funktion berechnet, wobei φ für eine mögliche Abweichung in der Zeit steht.

$$P_{(x=\varphi)} = \frac{e^{\frac{\varphi}{\sum T}} - e^{-\frac{\varphi}{\sum T}}}{e^{\frac{\varphi}{\sum T}} + e^{-\frac{\varphi}{\sum T}}}$$

Dabei veranschaulicht die **Abbildung 35** die mögliche Abweichung in der Zeit wobei auf der X-Achse das Verhältnis zwischen der abgewichenen Reisezeit (in Sekunden) zur Gesamt-Reisezeit angezeigt wird. Beispielsweise bedeutet der Wert 5.0 in **Abbildung 35**, dass die Wahrscheinlichkeit für eine Abweichung von 50% von der Gesamt-Reisezeit bei ca. 0,46 liegt.

Ähnlich wie bei der Abweichung in der Raumdimension, $P_{(x \leq \varphi | Y=1)}$ und $P_{(x \leq \varphi | Y=0)}$, kann die Zeitabweichung mittels Gaußscher Verteilung berechnet werden. Bezugnehmend auf die **Hypothese I** aus Kapitel 4.3 soll der Versuch des Benutzers, wieder zurück an die ursprüngliche Route zu gelangen, als ein echter Versuch die Abwei-

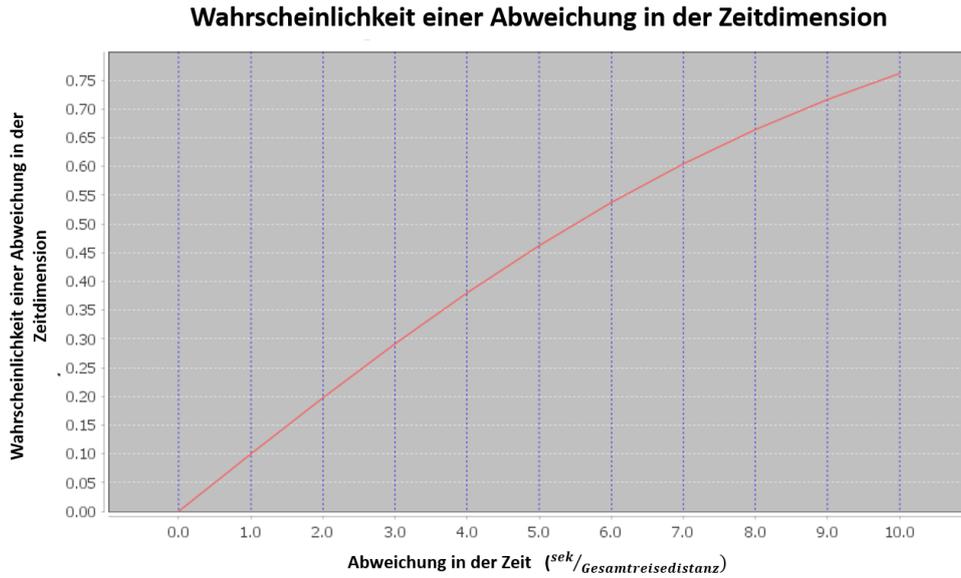


Abbildung 35: Wahrscheinlichkeit einer Abweichung in der Zeitdimension.

chung auszugleichen vom Algorithmus aufgefasst werden. Für jedes Mal, wenn der Benutzer sich der ursprünglichen Route r , nach einer erkannten Abweichung nähert, entwickelt sich die mögliche Abweichung φ folgendermaßen zu φ_r , wobei $P_{(\omega)}$ die Wahrscheinlichkeit der Abweichung in der Logik bezeichnet, welche im Kapitel 4.4.4 diskutiert wird. Dabei bezeichnet φ_0 die Abweichung am Startpunkt:

$$\varphi_1 = \prod P(x_t; \theta, \varphi, \omega | Y = 1)^0 * \varphi_0,$$

$$\varphi_2 = \prod P(x_t; \theta, \varphi, \omega | Y = 1)^1 * \varphi_1,$$

$$\varphi_r = \prod P(x_t; \theta, \varphi, \omega | Y = 1)^{r-1} * \varphi_{r-1},$$

$$\varphi_1 \varphi_2 \dots \varphi_r = \prod P(x_t; \theta, \varphi, \omega | Y = 1)^{0+1+\dots+r-1} * \varphi_0 \varphi_1 \dots \varphi_{r-1}.$$

$$\varphi_r = \varphi_0 \prod P(x_t; \theta, \varphi, \omega | Y = 1)^{\frac{r(r-1)}{2}}.$$

4.4.4 Deviationen in der logischen Dimension

Es existiert jedoch noch eine dritte Dimension für Abweichungen in intNAVs welche in der Literatur bis jetzt einzigartig ist. Diese dient dazu die erkannten Raum- und Zeitdivergenzen logisch zu bestätigen oder zu revidieren. Die logische Dimension hilft also erkannte Abweichungen zu bewerten und gibt ein Indiz dafür, ob es sich dabei um eine beabsichtigte Abweichung oder um eine unbeabsichtigte Abweichung des Benutzers handelt. Die Erkennung einer Abweichung kann auch dazu beitragen, die Eventualität eines Risikos einer Abweichung zu erkennen, während die mögliche Deviation noch nicht offensichtlich ist.

Eingangs von diesem Kapitel wurden beabsichtigte und unbeabsichtigte Abweichungen des Benutzers diskutiert. Dabei wurde betont, dass insbesondere in einem intNAV die Unterscheidung wichtig ist, da dadurch unnötiges Nerven des Benutzers ausgeschlossen werden kann und der geringe Platz, welcher auf einem Smartphone zur Verfügung steht, dadurch am besten ausgenutzt werden kann.

Die Logik-Abweichung kann am Besten anhand des folgenden Beispiels erklärt werden:

Ein Benutzer soll für seine Strecke 20 Minuten und 10 km benötigen. Nach 20 Minuten ist der Benutzer zwar schon in der Nähe von dem Zielort, möchte aber noch einen Parkplatz finden, deshalb fährt dieser noch weitere 5 Minuten und 2km, bis ein Parkplatz gefunden wurde.

*In diesem Fall wäre es sehr **nervig**, wenn der Benutzer ständig darüber in Kenntnis gesetzt wird, dass dieser abgewichen ist.*

*Hier greift nun die Abweichung in der Logik, um den **Einfluss der potentiellen Raum-/Zeit-Abweichung** in solchen Szenarien so stark wie möglich zu **reduzieren**.*

Dabei gehen die Raum-/Zeit-Abweichungen mit der Logik-Abweichung „Hand in Hand“, denn die Logik-Abweichung basiert auf den Raum-/Zeit-Abweichungen. Die gesamte Dimension der Abweichung in der Logik besteht aus einem positiven, negativen und neutralem logischen Schlussfolgern („Reasoning“). Zwar wurde eine Abweichung in der Raum-/Zeit-Dimension festgestellt aber anhand des positiven, negativen und neutralem logischen Schlussfolgern keine Abweichung in der Logik eingetreten ist. Anhand von nachfolgenden Beispielszenarien werden die drei genannten Komponenten erläutert und die entsprechenden Formeln eingeführt.

Positives logisches Schlussfolgern

In der Nähe des Zielortes

Besuchte POIs in der Nähe

Negatives logisches Schlussfolgern

Divergenz zwischen gesetztem Ort und aktuellem Ort

Anormaler Bewegungsverlauf

Neutrales logisches Schlussfolgern

Berechnung des Geschwindigkeitsvektor

Diese logischen Komponenten unterstützen oder widerlegen eine erkannte mögliche Abweichung, welche in den beiden anderen Dimensionen errechnet wurde. Das Resultat der logischen Komponenten wird als zusätzlicher Input zu dem Bayes'schem Netzwerk herangezogen, dargestellt als eine Wahrscheinlichkeit der Abweichung in der logischen Dimension $P_{(x=\omega)}$.

4.4.4.1 POSITIVES LOGISCHES SCHLIEßEN

Bei einem positiven logischen Schlussfolgern haben die beiden Dimensionen, Zeit und Raum, eine Abweichung erkannt, wohingegen die Logik **keine** Abweichung erkannt hat. Es gibt eine Vielzahl von Situationen, welche dafür sprechen, dass der Benutzer abgewichen ist, obwohl er es nicht ist. Wenn ein Benutzer beispielsweise einen Parkplatz sucht, sieht es im ersten Moment so aus, als wäre der Benutzer von der Route abgewichen. In solchen

Fällen ist die Wahrscheinlichkeit für eine Abweichung in der Raumdimension groß und je länger der Benutzer diesen Parkplatz sucht auch in der Zeitdimension. Das logische Schlussfolgern hilft nun, diese mögliche Abweichung zu „glätten“.

Dabei bezeichnet L_r den linearen Abstand zwischen der aktuellen Position und dem Zielort. L bezeichnet den linearen Abstand zwischen der aktuellen Position und dem Startpunkt. Der Parameter R ist der Activation Threshold und dient der Lokalisierung des Benutzers. Die Berechnung von $P_{(i_0)}$ wird aktiviert, sobald sich der Benutzer innerhalb eines Radius R , von dem Zielort entfernt, befindet. Der Relaxations-Coefficient C kann durch Training festgestellt werden um das Gewicht von R zu beschränken.

Daraus ergibt sich folgende Formel:

$$P_{(i_0)} = C \frac{L - L_r}{L}, L_r \in (0, R)$$

4.4.4.2 NEGATIVES LOGISCHES SCHLIEßEN

Die Idee hinter dem Negativen Schließen ist, dass zwar eine Abweichung (in der Zeit- oder Raum-Dimension) erkannt wurde, es jedoch keinen Sinn machen würde, diese dem Benutzer aus **logischer** Sicht mitzuteilen. Wenn der Benutzer vor dem Zeitplan von der Soll-Route abgewichen ist, macht es nicht viel Sinn, dies bereits als Abweichung zu deklarieren. Um eine mögliche Abweichung in der Raum-Dimension in diesem Fall zu kompensieren, wird $P_{(i_1)}$ berechnet. Dabei definiert S den aktuellen Zeitpunkt vom Start, und τ den geplanten Zeitpunkt (nämlich Soll-Zeitpunkt wann der Benutzer an dem Ort sein soll). Die Einheit in welchen diese Werte angegeben werden ist egal, sie müssen nur gleich und linear sein. In einfachen Fällen können Sekunden gewählt werden.

Die Wahrscheinlichkeit für so eine Abweichung kann folgendermaßen berechnet werden:

$$P_{(i_1)} = -\left(\frac{1}{2}\right)^{\frac{e\tau}{S}} + 1$$

Wenn der Benutzer ein anomales Bewegungsverhalten zeigt, beispielsweise der Benutzer stoppt, dreht um, dreht nochmal um, weist dies ein Anzeichen dafür auf, dass der Benutzer sich verirrt hat. Da dieses Benutzerverhalten keine Kontinuität aufweist, kann es nicht, wie die anderen Wahrscheinlichkeiten berechnet werden, sondern muss mittels Poisson-Verteilung erfolgen.

Die Vor- und Zurückbewegungen des Benutzers können als eine Binomialverteilung angesehen werden, wobei der Parameter n einen sehr hohen Wert und der Parameter p einen sehr niedrigen Wert einnimmt. Diese Binomialverteilung kann in etwa als eine Poisson-Verteilung modelliert werden, welches durch den Parameter λ festgelegt wird.

Die Häufigkeit, mit welcher der Benutzer stoppt oder zurückgeht wird durch N_S und N_B festgelegt. Das B steht hierbei für den englischen Ausdruck „backwards“ für rückwärts. Die Poisson-Verteilung folgt folgender Formel $P(x = i) = \frac{e^{-\lambda} \lambda^i}{i!}$, $i \in N$, wobei λ den erwarteten Wert als auch die Varianz darstellt.

Die Wahrscheinlichkeit, dass der Benutzer nun die Zielposition pünktlich erreicht, wobei dieser stoppt oder zurückgeht, kann folgendermaßen dargestellt werden:

$$P(x = N_S) = \frac{e^{-\lambda_1} \lambda_1^i}{i!}$$

$$P(x = N_B) = \frac{e^{-\lambda_2} \lambda_2^i}{i!}$$

$P_{(i_2)}$ ist dann der gewichtete Durchschnitt von $P(x = N_S)$ und $P(x = N_B)$.

Wenn nun mittels negativem Schließen der logischen Dimension solch ein Verhalten erkannt wird, kann frühzeitig auf eine mögliche unbeabsichtigte Abweichung hingewiesen werden.

Mit Hilfe der logischen Dimension wird zwar die Abweichung nicht weniger schwerwiegend, es wird jedoch der Grund für die Abweichung erkannt. Wie bereits beschrieben, soll je nach Art einer Abweichung der Benutzer anders darüber in Kenntnis gesetzt werden. Eine Abweichung verursacht durch Verirrung des Benutzers, soll dem Benutzer demnach intensiver auf seinem Smartphone mitgeteilt werden, als eine Abweichung verursacht durch Abbiegen in einen POI, beispielsweise Bäckerei. Bei solchen Abweichungen ist davon auszugehen, dass der Benutzer sich dieser Deviation bewusst ist und diese absichtlich in Kauf nimmt. Aus diesem Grund würde sich der Benutzer gestört fühlen, wenn die Applikation ihn intensiv auf diese Abweichungen hinweist. Besser wäre es, wenn das intNAV anhand der positiven logischen Schlussfolgerung die Intention des Benutzers erkennt und im Hintergrund Alternativen berechnet.

4.4.4.3 NEUTRALES LOGISCHES SCHLIEßEN

Ein Benutzer kann sich in der Nähe des Zielortes aufhalten, dennoch aufgrund von falschen Entscheidungen abgewichen sein. Beispielsweise ist der Benutzer an einer Kreuzung falsch abgebogen. Das Bewegungsverhalten des Benutzers sieht anders aus, als das prognostizierte. Dieser Vergleich kann mittels der Berechnung eines Geschwindigkeitsvektors erfolgen. Mit Hilfe des Geschwindigkeitsvektors können die Bewegungen des Benutzers aufgezeichnet und analysiert werden, sodass eine mögliche Abweichung in der Logik frühzeitig entdeckt werden kann.

Gegeben sei ein tatsächlicher Punkt $P_1(x_1, y_1)$ und ein Punkt der letzten aufgezeichneten Instanz $P_0(x_0, y_0)$, so kann der Geschwindigkeitsvektor der aktuellen Bewegung wie folgt beschrieben werden: $\vec{A} = (x_1 - x_0, y_1 - y_0)$. Abhängig von der vorgeschlagenen Route, kann der Geschwindigkeitsvektor der erwarteten Bewegung folgendermaßen definiert werden: $\vec{B} = (x_1^* - x_0^*, y_1^* - y_0^*)$. Das Skalarprodukt beider Geschwindigkeitsvektoren kann nachfolgend kalkuliert werden:

$$u = \cos(\theta) = \frac{\vec{A} * \vec{B}}{|\vec{A}| * |\vec{B}|} = \frac{(x_1 - x_0, y_1 - y_0) * (x_1^* - x_0^*, y_1^* - y_0^*)}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} * \sqrt{(x_1^* - x_0^*)^2 + (y_1^* - y_0^*)^2}}$$

Da $\cos(2k\pi) = 1$, $k \in \mathbb{Z}$ ist, welches eine ideale Bewegung des Benutzers darstellt, also dass sich der Benutzer in der richtigen Richtung, abhängig von der gewählten Route, befindet. Sobald es einen Winkel θ gibt, zwischen der aktuellen Route und der gewählten Route, gilt folgendes:

$$\cos(2k\pi) - \cos(\theta) = -2 \sin\left(\frac{2k\pi + \theta}{2}\right) \sin\left(\frac{2k\pi - \theta}{2}\right) = -2 \sin\left(k\pi + \frac{\theta}{2}\right) \sin\left(k\pi - \frac{\theta}{2}\right) = 2 \sin\left(\frac{\theta}{2}\right)^2 \gg 0.$$

Die Wahrscheinlichkeit, welche vom Geschwindigkeitsvektor abgeleitet wurde, kann mit Hilfe der Divergenz der Ist- und Soll-Bewegungsrichtung mittels Kumulieren der Winkel berechnet werden:

$$P_{(i_4)} = \frac{\sum_{i=0}^t (\cos(2k\pi) - u)}{t} = \frac{\sum_{i=0}^t (\cos(2k\pi) - \cos(\theta_t))}{t} = \frac{\sum_{i=0}^t 2 \sin\left(\frac{\theta_t}{2}\right)^2}{t}$$

Die Variable t stellt dabei die Bewegungszeit des Benutzers dar.

4.4.4.4 LOGISCHES SCHLIEßEN MIT HILFE EINES MULTI-SCHICHTEN NEURONALEN NETZWERKS (MNN)

Um das Vorgehen zur Erkennung einer Abweichung zu erklären, wird exemplarisch die Funktionsweise anhand eines 3-Schichten neuronalen Netzwerks erklärt. Dabei beinhaltet das Multischichten Neuronale Netz (MNN) fünf Eingangsneuronen (i_0, i_1 sind positiv, i_2, i_3 sind negativ und i_4 ist neutral) zwei versteckten Schichten und einem Ausgang.

Komponenten des Netzwerks

Die Eingangsneuronen können als i_0, i_1, \dots, i_4 beschrieben werden.

i_0 : Ausgang des **positiven Schlussfolgerns**. Benutzer ist in der Nähe des Ziels, erscheint jedoch als abgewichen. (Parkplatzbeispiel).

i_1 : Ausgang des **positiven Schlussfolgerns**. Benutzer besucht hier oft einen bestimmten Ort (Wahrscheinlichkeit hergeleitet aus den POIs in der Nähe)

i_2 : Ausgang des **negativen Schlussfolgerns**. Benutzer ist von der Soll-Route abgewichen; gleichzeitig ist er jedoch vor dem Zeitplan da. (Wahrscheinlichkeit abgeleitet aus der Divergenz zwischen Soll- und Ist-Position)

i_3 : Ausgang des **negativen Schlussfolgerns**. Benutzer zeigt anormale Bewegungsabläufe auf. (Wahrscheinlichkeit, hergeleitet aus den anormalen Bewegungsabläufen des Benutzers)

i_4 : Ausgang des **neutralen Schlussfolgerns**, abhängig von der Berechnung der Geschwindigkeitsvektoren. (Wahrscheinlichkeit, abgeleitet aus dem Geschwindigkeitsvektor)

Die Eingangsneuronen sind jeweils mit den versteckten Schichten (vgl. l_0 und l_1 in **Abbildung 36**) verbunden, deren adaptive Gewichte werden mit $w_{00}, w_{01}, w_{10}, w_{11}, \dots, w_{40}, w_{41}$ im Schaubild bezeichnet. Die Gewichte w_{k0} ($k \in \mathbb{N}$), sind mit dem Knoten l_0 verbunden, welches als positive Aktivierungseinheit bezeichnet wird, während die Gewichte w_{k1} ($k \in \mathbb{N}$), mit dem Knoten der negativen Aktivierungseinheit l_1 verbunden sind. Die

adaptiven Gewichte, welche die Verbindung zwischen den versteckten Schichten l_0, l_1 und dem Ausgang o , darstellen, werden als w_{l_0} und w_{l_1} bezeichnet.

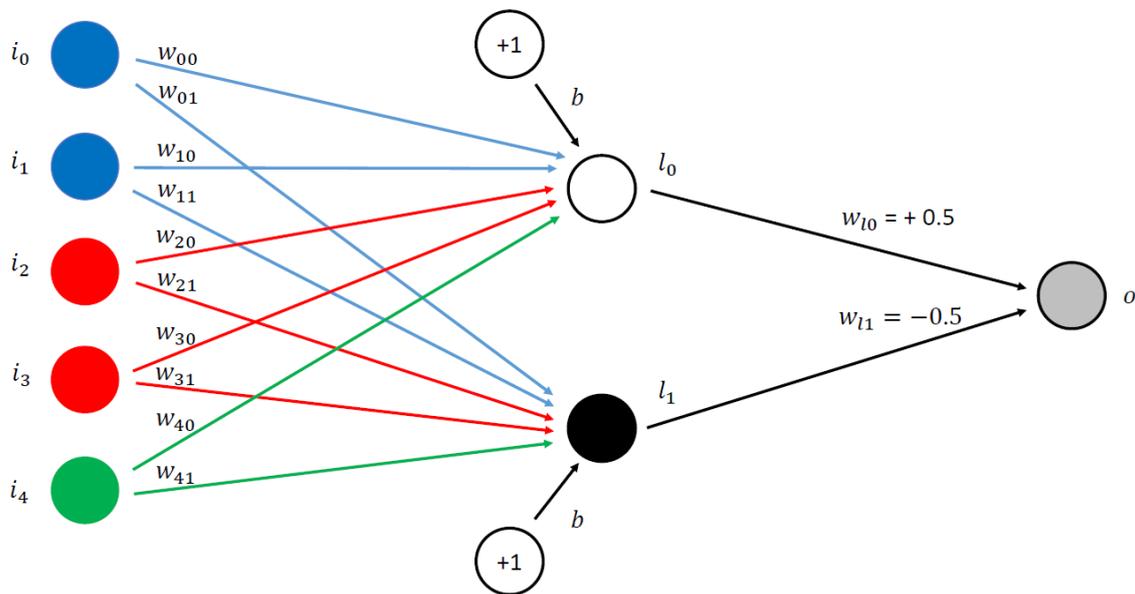


Abbildung 36: Schlussfolgerung basierend auf einem Multi-Schichten Neuronales Netzwerk

Initialisierung der Gewichte

Eine Abweichung kann durch mehrere Faktoren beeinflusst werden und jeder Faktor hat einen Einfluss auf die gesamte Abweichung. Für die Erkennung einer Abweichung, wird die Gauß'sche Verteilung auf das logische Schließen angewendet.

Gegeben sei eine Abweichung in der logischen Dimension $x, x \sim N(\mu, \sigma^2)$ und $Y = 0$ oder $Y = 1$, wenn eine Abweichung existiert.

Die Wahrscheinlichkeit, dass eine Abweichung in der logischen Dimension vorliegt, kann nun folgendermaßen berechnet werden:

$$P(x|Y = 1) = F(x; \mu, \sigma) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^x e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx$$

$$P(x|Y = 0) = 1 - F(x; \mu, \sigma) = 1 - \int_{-\infty}^x \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 1 - \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^x e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx$$

Bevor das Benutzerverhalten gelernt werden kann, weiß das Netzwerk nichts über die erste Eingabe x , aus diesem Grund, werden die initialen Werte der Gewichte mit $x = 0,5$ belegt. Die initialen Gewichte w_{l_0} und w_{l_1} sind somit $\pm 0,5$.

Die initialen Gewichte zwischen den Neuronen des positiven Schließens i_0 und der positiven Aktivierungseinheit l_0 werden folgendermaßen berechnet:

$$w_{00} = P(i_1 = 0,5|Y = 0) = 1 - F(i_1; \mu_1, \sigma_1) = 1 - \frac{1}{\sqrt{2\pi}\sigma_1} \int_{-\infty}^{0,5} e^{\left(-\frac{(0,5-\mu_1)^2}{2\sigma_1^2}\right)} dx$$

Analog verhält es sich mit dem Initialisieren des Gewichtes zwischen der Logik des positiven Schließens i_0 und der negativen Aktivierungseinheit l_1 :

$$w_{01} = P(i_1 = 0,5|Y = 1) = F(i_1; \mu_1, \sigma_1) = \frac{1}{\sqrt{2\pi}\sigma_1} \int_{-\infty}^{0,5} e^{\left(-\frac{(0,5-\mu_1)^2}{2\sigma_1^2}\right)} dx$$

Die Gewichte jeder Verbindung zur versteckten Schicht der positiven Aktivierungseinheit l_0 werden folgendermaßen berechnet:

$$w = P(x|Y = 0) = P(x = 0,5|Y = 0) = 1 - F(x; \mu, \sigma) = 1 - \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^x e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)} dx$$

Wohingegen die Gewichte der Verbindungen zur versteckten Schicht der negativen Aktivierungseinheit l_1 folgendermaßen berechnet werden:

$$w = P(x|Y = 1) = P(x = 0,5|Y = 1) = F(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^x e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)} dx$$

Allgemein kann somit die Initialisierung der Gewichte zwischen den Eingangsneuronen und den versteckten Schichten eines MNN folgendermaßen berechnet werden:

$$w_{ij} = P(x_{i+1} = 0,5|Y = j) = \sin\left(\frac{(j+1)\pi}{2}\right) + \sin\left(\frac{(j-1)\pi}{2}\right) \frac{1}{\sqrt{2\pi}\sigma_{i+1}} \int_{-\infty}^{0,5} e^{\left(-\frac{(0,5-\mu_{i+1})^2}{2\sigma_{i+1}^2}\right)} dx, i \in \mathbb{N}, j \in \{0,1\}$$

Lernen zur Optimierung der Vorwärts- und Rückwärts Propagation

Nach der Initialisierung der adaptiven Gewichte, kann das Lernen innerhalb des MNN mittels Vorwärts- und Rückwärts Propagation erfolgen. Die Aktivierungsfunktion des Netzwerkes wird durch eine Sigmoid-Funktion

$S(x) = \frac{1}{1+e^{-x}}$ abgebildet, dessen Ableitung die Funktion $\frac{dS(x)}{dx} = S(x)(1 - S(x))$ darstellt.

Für das erste Update, wird die gewichtete Summe der Eingangsneuronen berechnet:

$$l_0 = w_{00}i_0 + w_{10}i_1 + \dots + w_{40}i_4$$

$$l_1 = w_{01}i_0 + w_{11}i_1 + \dots + w_{41}i_4$$

Dies kann mit Hilfe einer Matrix dargestellt werden:

$$L_j = W_{ij} * I_j, i \in \mathbb{N}, j \in \{0,1\}$$

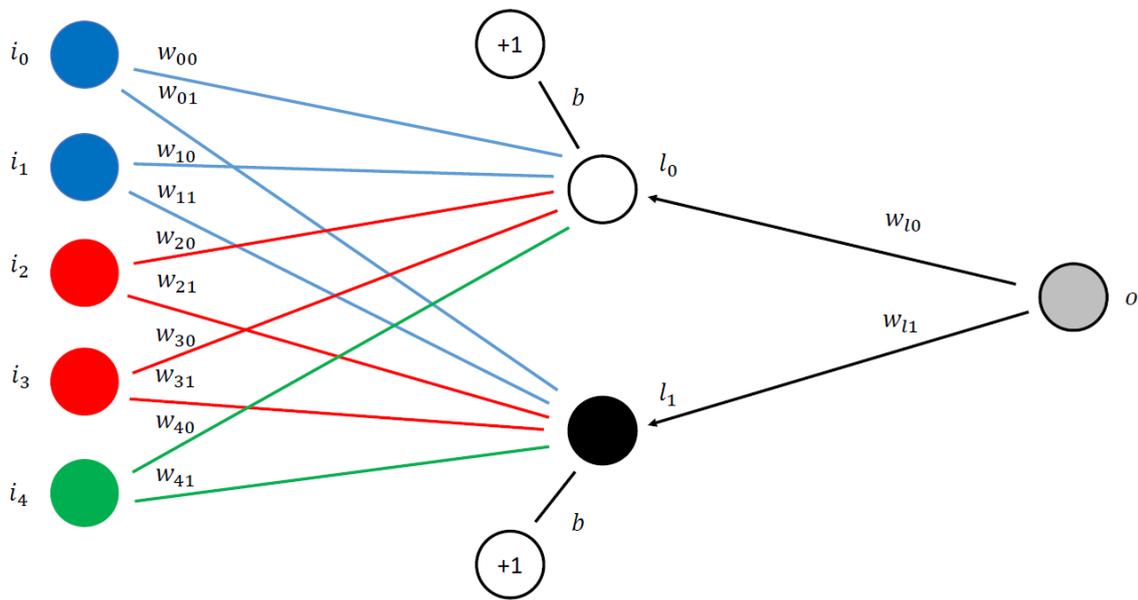


Abbildung 37: Lernen mit Hilfe der Rückwärts-Propagation

Aktiviert durch die Sigmoid-Funktion $S(x)$ und weiterer Vorwärts-Propagation zum Ausgangsknoten o , entsteht ein Fehler, welcher mittels $E = (y - y_0)^2$ berechnet wird, wobei y_0 die erwartete Eingangsgröße ist.

Während der Rückwärts-Propagation, wird der Fehler mittels $-(y - y_0) * \frac{dS(x)}{dx} = -(y - y_0) * s(x)(1 - S(x))$ ermittelt. Das Aktualisieren der Gewichte endet, sobald der Fehler eine Konvergenz erreicht.

4.5 PROGNOSE MÖGLICHER DEVIATION

Sobald das Ergebnis der Wahrscheinlichkeitsberechnungen aller drei Dimensionen ($P_{(x=\theta)}$, $P_{(x=\varphi)}$ und $P_{(x=\omega)}$) berechnet wurde, kann die Vorhersagephase parallel dazu gestartet werden. Bezogen auf die **Hypothesen I, II** und **III** ist nicht zu erwarten, dass der Benutzer sein Ziel sprunghaft ändert, sondern sich mit seiner „normalen“ Geschwindigkeit bewegt.

Die Prognose einer möglichen Deviation kann mittels exponentieller Glättung nach *Brown et al.* (Brown, et al., 1961) erfolgen, da die Methode auf Zeitreihen angewendet werden kann um statistische Prognosen zu erhalten. Exponentielle Glättung beschreibt, dass Zeitreihen sich rational bewegen und vergangene Zeitreihen einen bestimmten Einfluss auf zukünftige Zeitreihen besitzen, wenn geeignete Gewichte ausgewählt und angewandt werden. Die Kontinuität und Regularität der Datenserie ist dabei eine Voraussetzung. Da die Daten über die Zeitreihen der Deviation in allen drei Dimensionen die Vorbedingungen erfüllen und eine mögliche Deviation des aktuellen Zeitpunkts unzertrennlich vom letzten Zeitpunkt abhängt, stellt eine **doppelte exponentielle Glättung** das ideale Werkzeug dar um eine mögliche Deviation zu prognostizieren.

Gegeben sei eine Beobachtung unabhängig von der Dimension einer möglichen Abweichung als $\{y_k\}$, $y_k = P(x_k)$, $x_k \in \{\theta, \varphi, \omega\}$, $k \in \mathbb{N}^*$.

Die geglättete Statistik (engl.: „smoothed statistic“) $S_t^{(1)}$ des Zeitpunkts t , wird folgendermaßen definiert:

$$S_t^{(1)} = a * y_t + (1 - a) * S_{t-1}^{(1)}$$

Die doppelte geglättete Statistik (engl.: „double smoothed statistic“) $S_t^{(2)}$ zum Zeitpunkt t , wird demnach folgendermaßen berechnet:

$$S_t^{(2)} = a * S_t^{(1)} + (1 - a) * S_{t-1}^{(2)}$$

Wobei $t \in \mathbb{N}^*$, $S_0^{(2)} = S_0^{(1)} = y_1$.

Der Glättungsfaktor a , welcher zwischen (0,1) liegt wird basierend auf Erfahrungswerten zwischen 0,3 bis 0,7 gewählt. Sobald ein Benutzer von der vorgeschlagenen Route abweicht, wächst das Ergebnis der Deviation $\prod P(x_k; \theta, \varphi, \omega | Y = 0)$, wenn die Wahrscheinlichkeit einer möglichen Deviation $P(x_k)$ in den Dimensionen stark ansteigt. Wenn Personen sich an die vorgeschlagene Route halten, bleibt die potentielle Deviation in den Dimensionen $P(x_k)$ stabil und die totale Deviation $\prod P(x_k; \theta, \varphi, \omega | Y = 0)$ befindet sich auf einem niedrigen Level. Dies entspricht den Anforderungen an den Glättungsfaktor a . Da sich $\prod P(x_k; \theta, \varphi, \omega | Y = 0)$ im Bereich zwischen [0,1] befindet, kann der Glättungsfaktor a dynamisch gewählt werden. Um eine mögliche Abweichung festzustellen, wird die Wahrscheinlichkeit der Parameter mit der Voraussetzung berechnet, dass eine Abweichung stattgefunden hat, also a wird für $Y = 1$ berechnet. Auf der anderen Seite kann natürlich auch die Wahrscheinlichkeit der Parameter mit der Voraussetzung berechnet werden, dass die Abweichung nicht stattgefunden hat (also wird der Wert für $Y = 0$ niedrig sein.) In diesem Fall interessiert uns die zweite Wahrscheinlichkeit nicht.

Daher kann die doppelte exponentielle Glättung folgendermaßen erfolgen, bei dem $Y = 1$ verwendet wird.

$$S_0^{(2)} = S_0^{(1)} = y_1,$$

$$\begin{aligned} S_1^{(1)} &= a * y_1 + (1 - a) * S_0^{(2)} \\ &= \prod P(x_1; \theta, \varphi, \omega | Y = 1) * P(x_1) + (1 - \prod P(x_1; \theta, \varphi, \omega | Y = 1)) * P(x_1) = P(x_1) \end{aligned}$$

$$\begin{aligned} S_1^{(2)} &= a * S_1^{(1)} + (1 - a) * S_0^{(2)} \\ &= \prod P(x_1; \theta, \varphi, \omega | Y = 1) * P(x_1) + (1 - \prod P(x_1; \theta, \varphi, \omega | Y = 1)) * P(x_1) = P(x_1) \end{aligned}$$

$$\begin{aligned} S_2^{(1)} &= a * y_2 + (1 - a) * S_1^{(2)} \\ &= \prod P(x_2; \theta, \varphi, \omega | Y = 1) * P(x_2) + (1 - \prod P(x_2; \theta, \varphi, \omega | Y = 1)) * P(x_1) \\ &= \prod P(x_2; \theta, \varphi, \omega | Y = 1) * (P(x_2) - P(x_1)) + P(x_1) \end{aligned}$$

$$\begin{aligned}
S_2^{(2)} &= a * S_2^{(1)} + (1 - a) * S_1^{(2)} \\
&= \prod P(x_2; \theta, \varphi, \omega | Y = 1) * \left(\prod P(x_2; \theta, \varphi, \omega | Y = 1) \right) * (P(x_2) - P(x_1)) + P(x_1) \\
&+ \left(1 - \prod P(x_2; \theta, \varphi, \omega | Y = 1) \right) * P(x_1) = P(x_1) \\
&= \prod P(x_2; \theta, \varphi, \omega | Y = 1)^2 * (P(x_2) - P(x_1)) + P(x_1)
\end{aligned}$$

Diese doppelte exponentielle Glättung wird so lange ausgeführt, bis $S_t^{(1)}$ und $S_t^{(2)}$ ermittelt wurden.

Eine Prognose einer möglichen Deviation in jeglicher Dimension kann mit Hilfe der Parameter α_t und β_t folgendermaßen erfolgen:

$$\begin{cases} \alpha_t = 2S_t^{(1)} - S_t^{(2)} \\ \beta_t = \frac{a}{1-a} (S_t^{(1)} - S_t^{(2)}) = \frac{\prod P(x_t; \theta, \varphi, \omega | Y = 1)}{1 - \prod P(x_t; \theta, \varphi, \omega | Y = 1)} * (S_t^{(1)} - S_t^{(2)}) \end{cases}, t \in \mathbb{N}^*$$

Zu jedem Zeitpunkt in der Zukunft, kann die Wahrscheinlichkeit einer Deviation in jeglicher Dimension mittels

$$P(x_k) = P(x_{t_0+T}) = y_{t_0+T} = \alpha_{t_0} + \beta_{t_0} * T$$

ermittelt werden, wobei t_0 den aktuellen Zeitpunkt angibt und T das Zeitintervall darstellt.

4.6 ZUSAMMENFASSUNG

In diesem Kapitel wurde darauf hingewiesen, dass die Erkennung des Grundes für eine aktive Abweichung insofern relevant ist, als dass dadurch ein Benutzer mit Informationen nicht genervt wird und die begrenzte Fläche auf einem Smartphone optimal ausgenutzt werden kann. Es wurden Situationen vorgestellt, welche bei einer intermodalen Route zu Abweichungen führen können. Anschließend wurde ein Algorithmus präsentiert, welcher mit Hilfe der drei Dimensionen Raum, Zeit und Logik die Prognose ausgibt, ob die festgestellte Abweichung tatsächlich eine echte Abweichung ist und damit ob der Benutzer absichtlich oder unabsichtlich abgewichen ist. Dabei hilft die logische Dimension eine Abweichung in der Raum- und Zeit Dimension zu bestätigen oder zu revidieren. Mit Hilfe einer exponentiellen Glättung sollen dabei unbeabsichtigte von beabsichtigten Abweichungen abgegrenzt werden. Im nächsten Kapitel werden die beiden Algorithmen aus Kapitel 3 und Kapitel 4 zum entwickelten Prototypen intNAV zusammengefasst. Im Kapitel 6 wird die Funktionsweise mittels zwei Studien, quantitativ und qualitativ, getestet.

5 EVENTGETRIEBENE ARCHITEKTUR FÜR EIN INTNAV

Nachdem Lösungen für die beiden Grundpfeiler eines intNAVs, Transportmittelerkennung in Kapitel 3 und Prognose von Deviationen in Kapitel 4 diskutiert wurden, soll in diesem Kapitel die Architektur vorgestellt werden, welches eine performante Lösung für ein intNAV darstellt. Dabei wird das Konzept einer Eventgetriebenen Architektur im intNAV umgesetzt (2), nachdem die Grundlagen in (1) genannt wurden. Anschließend wird der Prototyp vorgestellt (3) und anhand eines Benchmark-Tests die Performanz und Skalierbarkeit dessen getestet (4). Abschließend werden die Ergebnisse des Kapitels zusammengefasst (5).

[Dieses Kapitel basiert auf der (1) **Veröffentlichung** „*Concept for an Intermodal Traveller Information System with Real-Time Data using Complex Event Processing*“ (Olga Birth, Michael Hoffmann, Markus Strassberger, Roman Roor, Johann Schlichter), September 2015 im IEEE 18th International Conference on Intelligent Transportation Systems - (ITSC 2015) pp.2293-2298 und der (2) **Masterarbeit** „*Concept and Prototype Development of an Event-Driven Architecture for an intermodal Route Guide*“ von Michael Hoffmann, Betreuung: Olga Birth, Aufgabensteller: Prof. Dr.-Ing. (Univ.) Eckehard Steinbach, abgenommen an der Fakultät für Electrical and Computer Engineering, Lehrstuhl für Medientechnologie am 18. Mai 2015]

Ein intNAV benötigt eine große Anzahl an unterschiedlichen Informationen um den Benutzer größtmöglich zu unterstützen. In den vorhergehenden Kapiteln 3 und 4 wurden diese Informationen als aktive und passive Komponenten bezeichnet. Die wichtigsten Größen hierbei sind die Statusinformationen eines Benutzers (Ist- und Sollzustand) aber auch die Statusinformation verschiedener Transportmittel (Ist- und Sollzustand) sowie die Abweichenerkennung und die Erkennung zukünftiger Zustände einer Route. Zusätzlich müssen einem intNAV diese Informationen in Echtzeit vorliegen. Im Gegenzug zu einem Autonavigationssystem muss das intNAV also mit einer viel größeren Anzahl an Eingangsfaktoren und Alternativen in Echtzeit umgehen können. Da, wie in Kapitel 3 beschrieben, das Smartphone als Verarbeitungseinheit dieser Informationen herangezogen wird, ist ein weiterer wichtiger Bestandteil des intNAVs eine performante Architektur, welche dafür sorgt, dass die Verarbeitung dieses großen Datenstroms in Echtzeit auf einem Smartphone in angemessener Zeit erfolgt. Um dies zu erreichen, wurde bei dem intNAV mit einer Eventgetriebenen Architektur⁶ (EDA) gearbeitet. Das vorliegende Kapitel widmet sich der Beschreibung und Anwendung der Eventgetriebenen Architektur für ein Smartphone sowie die Darstellung der Performanz Messung. Zunächst werden dabei die Grundlagen von EDA

⁶ Aus dem Englischen: Event-Driven Architecture (EDA)

erläutert um im nächsten Schritt die Anwendung von EDA auf intNAV zu demonstrieren. Abschließend wird eine Performanz Messung durchgeführt. Dabei sollen die folgenden drei zentralen Fragen beantwortet werden:

Kann eine Eventgetriebenen Architektur für intNAVs verwendet werden?

Was sind die Vor- und Nachteile einer Eventgetriebenen Architektur bei einem intNAV?

Wie skalierbar ist dabei das intNAV welches mit einer Eventgetriebenen Architektur arbeitet?

5.1 THEORETISCHER HINTERGRUND

Dieses Unterkapitel beschreibt die Eventgetriebene Architektur (EDA) und die damit verbundene Komplexe-Event Verarbeitung⁷(CEP) als eine wichtige Komponente davon. EDA findet dort Anwendung, wo eine große Datenmenge (Big Data) in Echtzeit verarbeitet werden soll und das System dabei skalierbar und agil arbeiten muss. Zunächst werden die charakteristischen Eigenschaften von EDA erklärt und dann die Wichtigkeit von CEP als Teil von EDA. Weiterhin werden die wichtigen Komponenten von CEP untersucht und mit anderen verfügbaren Technologien verglichen um Vor- und Nachteil von CEP zu beleuchten.

5.1.1 Eventgetriebene Architektur (EDA)

EDA ist eine Softwarearchitektur bei welcher einzelne Komponenten mit Hilfe von Events kommunizieren. Eine solche Eventkommunikation arbeitet sequentiell. Beispielsweise ist ein Klick mit der Maus ein Event, welcher ein anderes Event generiert, z.B. „Öffnen eines Ordners“. Die grundlegende Idee ist, dass ein großes Softwaresystem aus vielen kleinen Komponenten mit jeweils eigener Funktionalität besteht welche mit Hilfe von Events kommunizieren. Ein solches Event kann eine Nachricht sein, welche eine andere Komponente darüber informiert, dass eine bestimmte Aufgabe erledigt ist. Events sind sehr wichtig innerhalb einer EDA. Aus diesem Grund nimmt das Verarbeiten und Weiterleiten von Events eine wichtige Rolle innerhalb von EDAs ein. Dabei muss ein EDA in der Lage sein Events zu erkennen, zu verarbeiten und entsprechend auf die Events reagieren (Luckham, et al., 2008).

Eine **EDA** besteht aus folgenden vier Schichten:

Eventgenerator: diese Schicht ist verantwortlich für die Transformation eines jeden relevanten Ereignisses in einem standardisierten Event. Diese Ereignisse können beispielsweise Messungen von physischen Sensoren sein.

Eventkanal: der Kanal transportiert die Events vom Generator zur Eventverarbeitungseinheit, z.B. könnte diese eine Internetverbindung sein.

Eventverarbeitungseinheit⁸ (EPE): in dieser Einheit werden die Events identifiziert und eine entsprechende Reaktion ausgeführt.

„Downstream“ Eventgetriebene Aktivität: diese wird als direkte Konsequenz eines erkannten Events in der Eventverarbeitungseinheit gestartet. Als Resultat können zum Beispiel weitere Anwendungen gestartet werden.

⁷ Aus dem Englischen: Complex Event Processing (CEP)

⁸ Aus dem Englischen: Event processing engine (EPE)

Events treten in einem sogenannten Eventstrom und in unterschiedlichen Abstraktionsschichten auf (vgl. **Abbildung 38**). Kontinuierliche und häufige Events, wie beispielsweise Sensordaten, sind dabei Events der niedrigen Abstraktionsebene. In der höchsten Abstraktionsebene befinden sich komplexe Events, welche eine große Bedeutung für die Applikation besitzen. Wenn ein bestimmtes Muster auf der niedrigen Abstraktionsebene erkannt wurde, wird daraus ein neues komplexes Event auf der höchsten Ebene generiert. Somit beinhalten die komplexen Events die relevanteste Information – unwichtige Daten werden dabei auf der niedrigen Abstraktionsebene herausgefiltert.

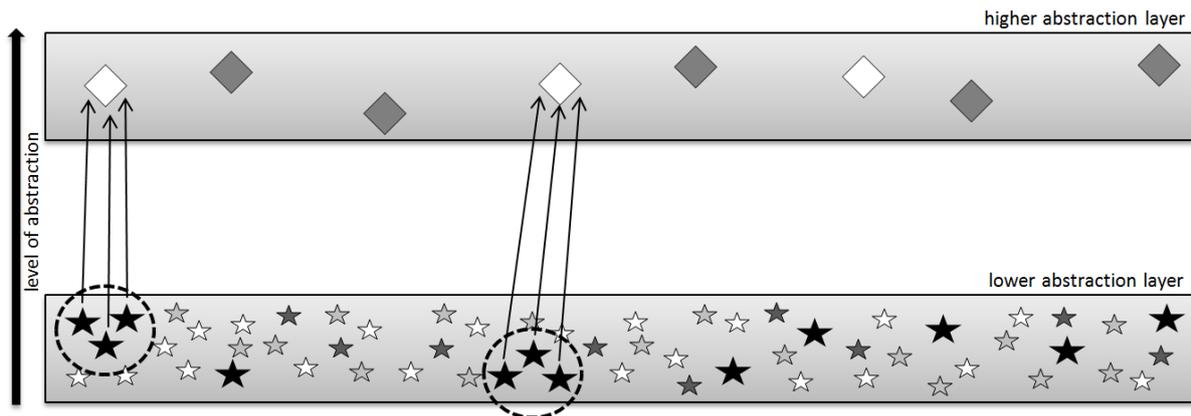


Abbildung 38: Events verschiedener Abstraktionslevel. Events auf der unteren Abstraktionsebene sind einfache Events. Komplexe Events können im höheren Abstraktionslevel gefunden werden. (nach (Burns, et al., 2010))

In **Abbildung 39** werden die logischen Schichten eines EDAs verdeutlicht. Dabei werden verschiedene Eventquellen als Sensoren oder weitere Informationssysteme illustriert. Die EPE erhält diese Ströme mit Hilfe von in-Adapttern. Dabei kann die Verarbeitung von Events aus mehreren CEPs bestehen welche miteinander kommunizieren. Die Out-Adapter senden dabei die generierten komplexen Events zu Anwendungen, Datenbanken, Services, etc.

Wie lassen sich nun aus einfachen Events komplexe Events bilden? Beispielsweise mit **Aggregation**. Dabei werden viele, wenig relevante, Events zu neuen Events mit höherer Signifikanz zusammengefasst. Die Signifikanz eines einzelnen Events hängt von dessen Granularität ab. Ein Beispiel soll dies verdeutlichen: Jede Anfrage eines Web Servers wird als ein Event im System registriert. Nun kann eine große Menge dieser Anfragen von verschiedenen IP Adressen in einer schnellen Abfolge erfolgen. Für sich alleine betrachtet, liefern diese Events keine bessere Information. Wenn jedoch der gesamte Anfragestrom betrachtet wird, dann wird klar, dass der Web Server ein Opfer einer Distributed Denial of Service (**DDoS**) Attacke war. Diese Attacke kann als ein neues Event auf der höheren Abstraktionsebene interpretiert werden.

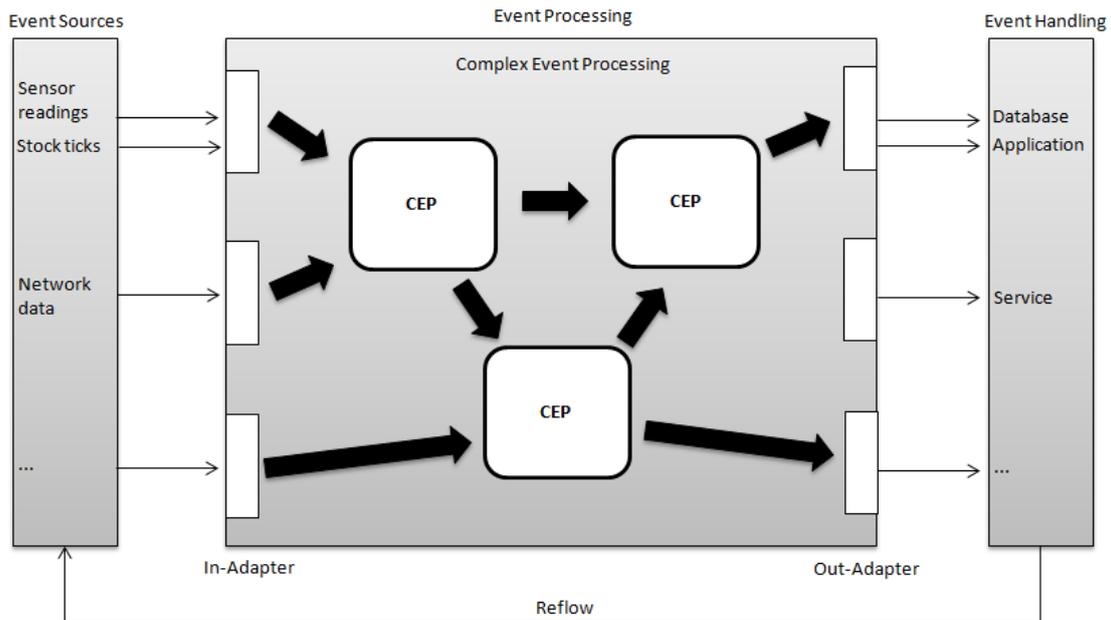


Abbildung 39: Logische Schichten eines typischen EDA: Die Eventverarbeitungsschicht (Event Processing Layer) beinhaltet einen oder mehrere CEPs und erhält Events aus unterschiedlichen Quellen. Die resultierenden komplexen Events werden weitergeleitet an die Event Handling Schicht um dort weitere Aktionen auszuführen (nach (Burns, et al., 2010)).

Eine weitere Möglichkeit, aus einfachen Events komplexe Events zu erhalten, liegt im **Filtern**. Dabei werden gleiche Events herausgefiltert, sodass Redundanzen vermieden werden können. Ein Beispiel ist der Radio-Frequency Identification (**RFID**)-Leser. Dieser generiert für jedes gelesene Element ein Event. Wenn nun der Leser mehrere Elemente vom gleichen RFID-Tag in einer sehr kurzen Zeit erhält, können diese duplizierten Elemente herausgefiltert werden und werden in der höheren Abstraktionsschicht nicht mehr verwendet.

Zusätzlich können mittels **Anreicherung** von Events mit zusätzlichen Daten komplexe Events geschaffen werden. Dazu soll wieder das Beispiel des RFID Lesers verwendet werden: die Events des RFID-Lesers enthalten nur die ID und einen Zeitstempel. Ein zusätzlicher Agent in der CEP kann weitere Informationen über den RFID-Label von der Datenbank beziehen. Die neue Information würde dann zu den Events hinzugefügt und die weiteren Agenten müssen diese Information nicht selbst beziehen.

Weitere Verarbeitungsmöglichkeiten von CEP können in der Literatur nachgeschlagen werden (Michelson, 2006).

5.1.2 Grundlagen von CEP

CEP ist ein integraler Bestandteil von EDAs. Etzion et al definierten das Basiskonzept und die Architektur von CEP (Etzion, et al., 2010). Ein Event kann dabei „alles sein was passiert, oder als passiert betrachtet werden.“, beispielsweise der Kauf einer Aktie (Luckham, 2001). CEP verwendet Techniken, Methoden und Werkzeuge um Events kontinuierlich und in Echtzeit zu verarbeiten. Filtern, Aggregieren und Anreichern einfacher Events um daraus komplexe Events zu erhalten mit wichtiger Kontextinformation. Die Börse ist eine der wichtigsten

Anwendungsfälle für CEPs um in Echtzeit und kontinuierlich den Aktienpreis zu analysieren und Trend frühzeitig zu erkennen.

5.1.3 Event

Die Basis von EDA und somit von CEP sind Events. Jedes Vorkommnis in einem technischen Kontext kann als Event repräsentiert werden, so lange es technisch erkennbar ist ((Burns, et al., 2010), Seite 108). Ein einzelnes Event ist nicht ausreichend um einen größeren Kontext oder zusätzlichen Nutzen darin zu sehen. Je mehr Events auftreten, desto besser können diese zu einem größeren Kontext zusammengefasst werden. Beispielsweise fallen folgende Events an:

ein Haus wird gebaut

würfeln

Ein Gefängnis wird besucht

jemand hat in der Lotterie gewonnen

Wenn nun die einzelnen Events betrachtet werden, wird mancher daraus nicht sofort auf das Spiel Monopoly schließen (inspiriert durch (Alves, et al., 2013), Seite 7). Wenn jedoch alle Events als Ganzes betrachtet werden, erkennt unser Gehirn den Zusammenhang zwischen den Events, welches ebenfalls die Aufgabe von CEP in EDA ist. Ein Event muss hierbei mindestens die folgende Information beinhalten, sodass es von der EVA verarbeitet werden kann: eindeutige Event ID, Zeitstempel, Quelle, Nutzdaten (Payload).

5.1.3.1 EVENTREGELN

Eventregeln müssen definiert werden, damit relevante Muster in Eventströmen erkannt werden können. Diese Muster können mit Hilfe sog. Event-Verarbeitungssprache⁹ (EPL) beschrieben werden und beinhalten einen Bedingungs- und Ausführungs-Teil (if - then). Der Bedingungs-Teil enthält einen Trigger für ein oder mehrere Muster, welche kombiniert werden können. Im Ausführungs-Teil wird ein Code definiert, welche ausgeführt wird, wenn eines der Muster des Bedingungs-Teils im Eventstrom erkannt wurde. Eine Kreation eines neuen Events oder Starten eines externen Service oder Applikation sind Beispiel eines solchen Ausführungs-Teils.

Luckham führte in seinem Buch eine EPL ein, die sog. RAPIDE-EPL (Luckham, 2001). Mit Hilfe von Operatoren wie and, or und -> können Eventmuster erkannt werden. Einige Beispiele zur Verwendung der Regeln:

when (A and C or D) then action_X: action_X wird ausgeführt wenn A und C oder D innerhalb eines Eventstroms sind.

when (B-> (A or C)) then action_Y: action_Y wird ausgeführt wenn A oder C auftreten nachdem B im Eventstrom aufgetreten ist.

Zurzeit gibt es keinen Standard, welcher eine formale Deklaration von Events, ihren Mustern oder Regeln liefert. Jedoch wurde einiges Richtung Standardisierung unternommen. Ein Beispiel ist das Event Processing Glossary (Luckham, et al., 2008) der Event Processing Society. Weiterhin können Events mittels Unified Modelling

⁹ Aus dem Englischen: Event Processing Language (EPL)

Language (**UML**) modelliert werden (Eckert, et al., 2009). Mishra et al stellten in ihrer Veröffentlichung eine Kombination von Oracle Continuous Query Language (**CQL**) mit Streambase StreamSWL vor um eine gemeinsame Sprache zu bilden (Fülöp, et al., 2010).

5.1.3.2 EVENTVERARBEITUNGS AGENT¹⁰ (**EPA**)

Ein EPA wird verwendet um kontinuierlich Events von Eventströmen zu erhalten und um diese an die Eventverarbeitungseinheit (EPE) zu schicken. Die EPE verwendet die vordefinierten Eventregeln und beobachtet damit den Eventstrom. Wenn ein Muster entdeckt wurde, wird der Ausführungsteil entsprechend ausgeführt.

Abbildung 40 verdeutlicht die Arbeitsweise von EPA.

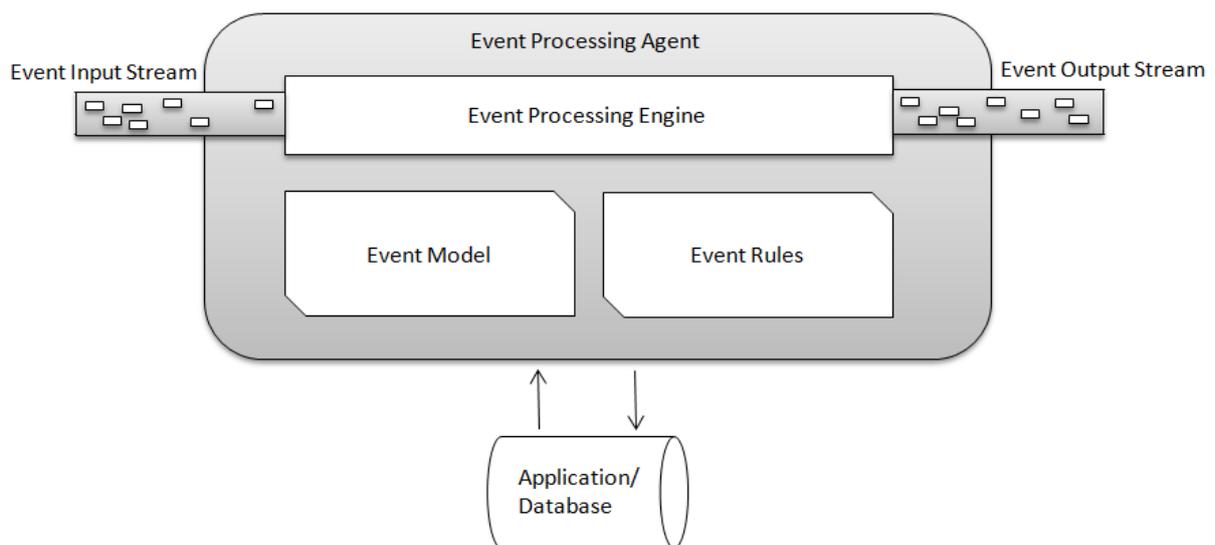


Abbildung 40: Elemente von EPA: Events werden durch die EPA mittels der Input und Output Ströme geschickt. Die Events werden mit Hilfe der sog. Eventmodelle (Event Models) definiert. Der Eventstrom wird mit den Eventregeln innerhalb der Eventverarbeitungseinheit (Event Processing Engine) verglichen. Die Events können um weitere Informationen aus der externen Datenbank angereichert werden (nach (Burns, et al., 2010))

5.1.3.3 EVENTVERARBEITUNGS NETZWERK¹¹ (**EPN**)

Viele miteinander verbunden EPAs bilden ein EPN. Komplexe Systeme werden oft in mehrere EPAs aufgeteilt. Ein erster EPA könnte eine einfache Aggregation oder Filtern von Events ausführen. Der daraus entstehende Strom wird dann als Eingangsgröße für den nächsten EPA verwendet. So behält das System eine klare Struktur und ist einfacher zu managen.

5.1.4 Verfügbare Software

Es gibt verschiedene kommerzielle und Open-Source Software zu Erstellung von eventgetriebenen Applikationen. Hier soll nun ein kurzer Abriss über die wichtigsten Produkte gegeben werden und diese miteinander verglichen werden um die beste CEP Software für intNAV zu finden.

¹⁰ Aus dem Englischen: Event Processing Agent (EPA)

¹¹ Aus dem Englischen: Event Processing Network (EPN)

IBM Active Middleware Technology (AMiT)

Active Middleware Technology (**AMiT**) von International Business Machines (IBM)¹² ist eine CEP Engine. AMiT basiert auf **IBM's** proprietärer WebSphere Brokering Technologie und trennt CEP in zwei Knoten: der IntelligentFilter Knoten wird als Filter verwendet welcher Situationen mit Hilfe von CEP erkennen kann und die Ergebnisse weiterleiten kann. Der SituationManager Knoten überwacht und informiert wenn solche Situationen ankommen. IBM Haifa Forschung wendete diese CEP Technologie in verschiedenen Industrien wie Banken, Versicherungen, Gesundheitswesen, Chemie und weiteren an (Magid, et al., 2010). Sie zeigten, dass die Technologie „hocheffizient und sehr erfolgreich“¹³ in verschiedenen Domains, Industrien und Applikationen ist. Durch die Trennung der EPE von dem Rest der Applikationslogik gewinnt der Abstraktionslevel an Agilität und Flexibilität. Da eine Vor- und Nachbearbeitung der Events notwendig ist, wird ein Netzwerk von EPAs vorgeschlagen.

ruleCore

ruleCore¹⁴ ist ein kommerzieller CEP Server welcher in der Lage ist, komplexe Eventstrukturen aus mehreren Eventströmen zu erkennen. Die Engine kann kontinuierliche Eventregeln auswerten, welche in einer Extensible Markup Language (**XML**)-basierten Eventregel Sprache namens Reakt definiert sind. Reakt definiert welche Situationen entdeckt werden sollen und welche nicht. Um ruleCore CEP Server verwenden zu können wird jedoch eine Lizenz benötigt.

Oracle CEP

Oracle CEP (früher bekannt als WebLogic Event Server)¹⁵ ist ein Java Server um hoch-performante und eventgetriebene Applikationen zu entwickeln. Diese Java Applikations Container basiert auf Equinox **OSGi** mit Oracle CEP Service Engine. Muster in mehreren Eventströmen können in Echtzeit mittels Oracle CQL erkannt werden. Mit Hilfe von JRockit Realtime Java Development Kit (JDK) sind „bis zu 1 Million Events pro Sekunde möglich bei gleichzeitig geringer durchschnittlicher Latenz“ (Corporation, 2008)¹⁶. Eclipse Entwickler können Oracle CEP Visualizer und Oracle CEP Integrated Development Environment (**IDE**) benutzen um EDA Applikationen zu entwickeln. Jedoch wird auch hierfür eine Lizenz benötigt.

StreamBase

TIBCO StreamBase® CEP Plattform ist ein „hoch-performantes System um schnell Applikationen zu erstellen, welche mit Echtzeitdaten arbeiten“¹⁷. Die Plattform besteht aus einer Entwicklungsumgebung, einem Event

¹² http://www.research.ibm.com/haifa/dept/services/papers/amt_fact_sheet.pdf

¹³ Aus dem Englischen: „highly effective and greatly successful“

¹⁴ <http://rulecore.com>

¹⁵ http://docs.oracle.com/cd/E16764_01/doc.1111/e14476/overview.htm

¹⁶ Aus dem Englischen: „up to 1 million events per second were possible while maintaining low average and peak latencies“

¹⁷ Aus dem Englischen: „a high-performance system for rapidly building applications that analyze and act on real-time streaming data“, <https://www.tibco.com/products/event-processing/complex-event-processing/streambase-complex-event-processing>

Server mit sehr geringer Latenz und hohem Durchsatz. Richard Tibbetts charakterisierte die Performanz und Skalierbarkeit von Handelssystemen mit Hilfe von TIBCo StreamBase® CEP Plattform durch Durchsatz, Latenz und Skalierbarkeit (Tibbetts, 2009). StreamBase kann bis zu 140.000 Events pro Sekunde verarbeiten mit einer Latenz von 86 µs auf einer Quad-Core Maschine. Mit Hilfe einer Eight-Core Maschine war das System in der Lage bis zu 245.000 Events pro Sekunde mit einer Latenz von 71 µs zu erreichen. Dadurch wird ein Skalierbarkeitsfaktor von 0.875 über Multi-Core Architekturen hinweg demonstriert. StreamBase gewann den Award Technology Pioneer vom World Economic Forum, für seine Leistung und Innovationen in der Eventverarbeitungs Technologie (Systems, 2010). Die Software kann für 90 Tage kostenlos danach nur mit Lizenz bezogen werden.

Esper

Esper ist eine der bekanntesten CEP Software, da es Open-Source, sowohl für Java als auch für C# verfügbar ist und eine verständliche und beliebte EPL hat. Es existiert eine sehr detaillierte Dokumentation¹⁸ sowie eine inoffizielle Schnittstelle für Android Smartphones, genannt Asper¹⁹. In **Abbildung 41** wird die generelle Architektur von Esper vorgestellt. Es kann für Nachrichten benutzt werden, welche in Echtzeit auftreten sowie für historische Daten. Esper verwendet seine eigene EPL und hat folgende Vorteile:

- Skalierbarkeit
- Berechnungen im Speicher
- SQL-Standard
- Geringe Latenz
- Echtzeit oder historische Daten



Abbildung 41: (N)Esper Architektur (Inc., 2015)

Esper verwendet die GNU General Public Licence (**GPL**) 2.0 Lizenz welche es erlaubt, dass die Software und dadurch auch die EPL von der Community kontinuierlich verbessert wird. Zusätzlich kann die Software umsonst

¹⁸ www.espertech.com/esper/documentation.php

¹⁹ <https://github.com/mobile-event-processin/Asper>

verwendet werden. Da Esper in Java bzw. in .NET geschrieben ist kann es in jede existierende Java- oder .NET-basierte Applikation integriert werden. Ein weiterer Vorteil ist, dass die Komponente unabhängig in einer Entwicklungsumgebung ausgeführt werden kann, welches die Entwicklung das Testen komfortabel gestalten. Somit gibt es keine Events welche zu einem Server geschickt werden müssen. Dadurch steigt die Performanz und Latenz von Anwendungen.

Esper bietet die Repräsentation von Events als Plain Old Java Object (**POJO**), Map oder XML an. Eine detaillierte Dokumentation mit über 700 Seiten und einer Menge an Beispielen ist verfügbar. Die EPL ist sehr ähnlich zur SQL Syntax. Ströme (EPL) sind das Äquivalent zu Tabellen/tables (SQL), Events (EPL) sind äquivalent zu Einträgen/record (SQL) und Eventattribute (EPL) sind einzelne Zellen/field in einem Eintrag (SQL). Esper besitzt kein Graphical User Interface (GUI) und kann nur mittels die Application Programming Interface (API) angesprochen werden.

Für die vorliegende Thesis wurde Esper als CEP Komponente ausgewählt. Die detaillierte Dokumentation, die verständliche EPL, die freie Verwendung basierend auf GNU GPL 2.0 Lizenz und viele verfügbare detaillierte Beispiele sind die größten Vorteile, weshalb Esper eine gute Wahl für CEP ist. Der Hauptgrund jedoch ist Asper, die Android Schnittstelle von Esper, welche aktuell die einzige verfügbare CEP Komponente für Android ist. Zusammengefasst ist Esper (respektive Asper) aktuell die beste Wahl für mobiles CEP.

5.1.5 Esper Details

In diesem Abschnitt werden die Grundlagen für die Verwendung von Esper gezeigt, welche benötigt werden um die Rest der vorliegenden Arbeit zu verstehen. Detaillierte Informationen können zusätzlich in der offiziellen Esper Dokumentation nachvollzogen werden (Inc., 2017).

Aktualisiere Listener in Esper

Der erste Schritt mit Hilfe der Esper API ist die Konfiguration und die Erstellung einer Instanz. Danach sollen die Statements registriert werden und einer oder mehrere Listener müssen zu den kreierte Statement(s) assoziiert werden. Folgender Code zeigt die Konfiguration des Esper Engine, die Registrierung von Statements und die Anfügen von einem Listener:

```
//Engine Configuration, add package including event classes
Configuration engineConfig = new Configuration();
engineConfig.addEventTypeAutoName(de.myapplication.cep.events.walking);
//Get engine instance
EPServiceProvider epService =
EPServiceProviderManager.getDefaultProvider(engineConfig);
//Register statement
String statement = „select * from GPSEvent“;
EPStatement statement = epService.getEPAdministrator().createEQL(statement);
//Attach a listener
Statement.addListener(new SampleListener());
```

Sobald neue Ergebnisse für ein Statement verfügbar sind, ruft die engine alle Listener Klassen welche mit diesem Statement assoziiert sind, auf. Die Events werden als EventBean Instanz gekapselt um so Eventeigenschaften und unterliegende Eventklassen besser abfragen zu können. Wenn ein Event ein Datenfenster betritt, erkennt die Engine dieses als newEvents und wenn es dieses Fenster verlässt als oldEvents. Eine Beispielimplementierung um einen Listener zu aktualisieren könnte so aussehen:

```
public class SampleListener implements UpdateListener{
publicvoid update(EventBean[] newEvents, EventBean[] oldEvents)
{
GPSEvent gpsEvent = (GPSEvent) newEvents[0].getUnderlying();
String latitude = (String) newEvents[0].get(„latitude“);
String longitude = (String) newEvents[0].get(„longitude“);
}
}
```

Eventregeln in Esper

Esper verwendet zwei Arten von Abfragen: Eventstromverarbeitung (ESP)- und CEP-Abfragen. ESP Abfragen sind zum Beispiel einfach Events, Events in einem gleitenden Fenster (sliding window) oder einem Fenster fester Größe. Nehmen wir zum Beispiel folgendes Statement:

```
select * from Withdrawal
```

Mit Hilfe dieser Statements werden alle Geldabhebe-Events ausgewählt. Alle update Listeners sind einbezogen jedes Mal wenn die CEP Engine ein Geldabhebe- Event prozessiert. Das neue Event wird dann an jeden Statement Listener weitergereicht. **Abbildung 42** zeigt das Auftreten von Abhebe Events (1-6) über die Zeit. Die Withdrawal Anzahl ist dabei die Zahl in Klammern. Das Beispiel resultiert nur in neuen Events und keine alten Events werden durch die Engine an den Statement's Listener geposted.

Um nur die letzten N events in der engine zu behalten kann eine Länge definiert werden (die Länge des Fensters). Das folgende Statement verwendet ein solches Statement:

```
select * from Withdrawal.win:length(N)
```

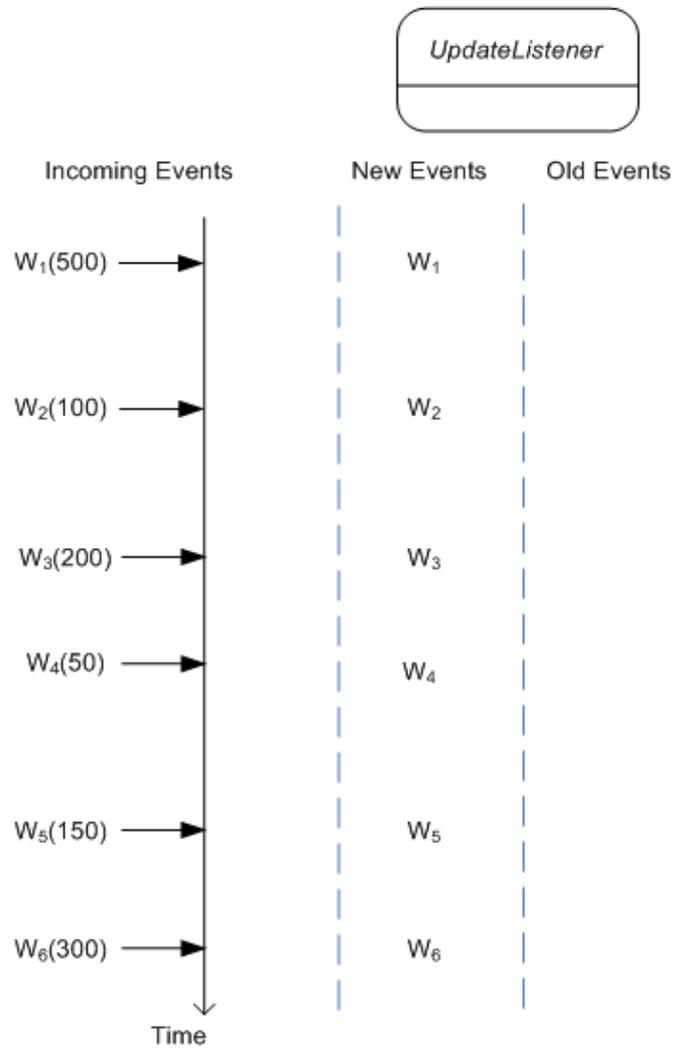


Abbildung 42: Beispiel für ein einfaches Statement `select * from Withdrawal`. Die linke Zeitleiste zeigt eine Serie von Abhebe-Events 1 bis 6 an, welche über die Zeit anfallen. Die Höhe die abgehoben wird, steht dabei jeweils in runden Klammern. Die zwei Spalten auf der rechten Seite zeigen, dass nur neue Events und keine alten Events berücksichtigt werden ((Inc., 2017), Seite 38).

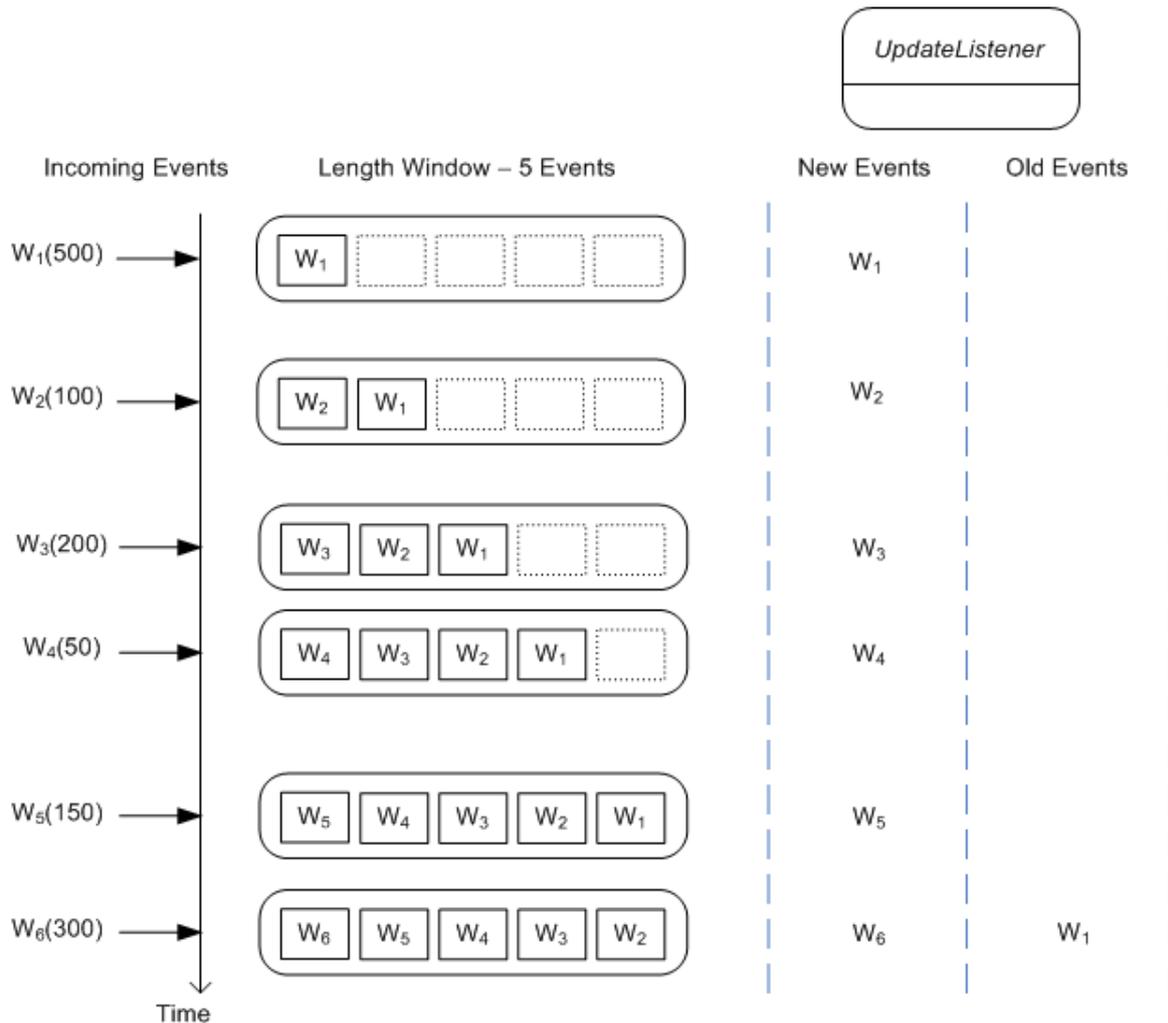


Abbildung 43: Beispiel für das Statement `select * from Withdrawal.win:length(N)`. Alle ankommenden Events werden als neue Events auf dem Listener berücksichtigt. Wenn das Event W₁ das Fenster aufgrund der Ankunft von W₆ verlässt, wird es als altes Event auf dem Listener eingefügt ((Inc., 2017), Seite 39).

Die Länge des Fensters in diesem Beispiel hat eine Größe von $N=5$ Events. Alle Geldabhebe-Events bedienen die CEP Engine. Wenn sich mehr als fünf Events in dem Fenster befinden und ein neues Event das Fenster betreten möchte, wird das älteste Event aus dem Fenster geschoben. Alle Events, welche das Fenster betreten werden als neue Events bezeichnet, alle die das Fenster verlassen, als alte Events. **Abbildung 43** zeigt wie sich der Inhalt des Fensters über die Zeit verändert und welche Events als neue und welche als alte markiert werden.

Das folgende Statement beinhaltet ein Zeitfenster um den spezifizierten Zeitintervall basierend auf der Systemzeit zu erweitern.

```
select * from Withdrawal.win:time(4 sec)
```

Abbildung 44 zeigt ein Diagramm welches u einer bestimmten Zeit t startet. Es veranschaulicht die Inhalte der Zeitfenster um $t+4$ und $t+5$ Sekunden, usw.

Folgende Aktivitäten finden statt:

- t+4: event W_1 erreicht das Zeitfenster. Das neue Event wird von dem System an den update Listener gemeldet.
- t+5: event W_2 erreicht das Zeitfenster. Das neue Event wird von dem System an den update Listener gemeldet.
- t+6: event W_3 erreicht das Zeitfenster. Das neue Event wird von dem System an den update Listener gemeldet.
- t+8: event W_1 verlässt das Zeitfenster. Das Event wird als altes Event an den Update Listener gemeldet.

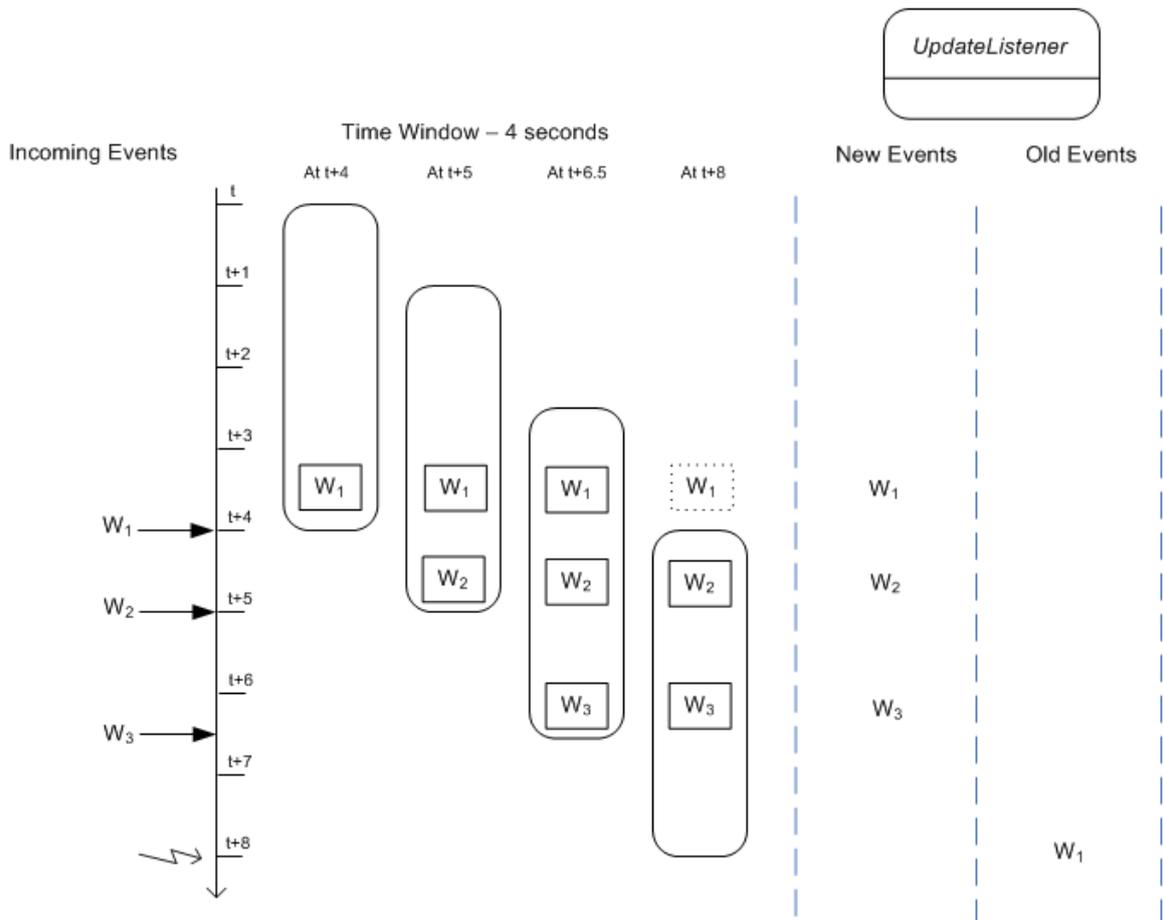


Abbildung 44: Beispiel des Statements `select * from Withdrawal.win:time(4sec)`. Das Diagramm startet zu einer bestimmten Zeit t und zeigt die Inhalte um $t+4$, $t+5$ usw. Das Event W_1 wird beim Listener um $t+4$ als neues Event gemeldet. Wenn $t+8$ eintritt, verlässt dieses Event das Fenster und wird als als erstes old Event zum Update Listener gemeldet ((Inc., 2017), Seite 43)

CEP Abfragen definieren Muster mit welchen die Eventströme überwacht werden um komplexe Events zu ermitteln. Muster werden einem Schlüsselwort definiert. Folgende Schlüsselwörter sind möglich:

every: geeignet für Wiederholungen

and, or, not: logische Operatoren

„->“ : bedingter Operator

timer:within: Mustererkennung mit Zeitrestriktion. Wenn ein bestimmtes Muster nicht innerhalb der Zeit erkannt wird, liefert dieser Ausdruck „false“.

timer:interval, timer:at: zeigt „true“, wenn eine bestimmte Zeit in Millisekunden abgelaufen ist. Der Ausdruck `timer:at` zeigt „true“ nach einer bestimmten Zeit.

Ein Beispiel für einen Eventstrom besteht aus folgenden Events:

A1 B1 C1 B2 A2 D1 A3 B3 E1 A4 F1 B4.

Wenn das Muster [every A -> B] angewandt wird, würde folgendes herauskommen:

A1, B1, A2, B3, A3, B3, A4, B4.

Wenn jedoch das Muster folgendermaßen definiert wäre [every (A -> B)], würde das Resultat folgendermaßen aussehen:

A1, B1, A2, B3, A4, B4.

EPL Abfragen erhalten die Information von einem oder mehrere Eventströme. EPL verwendet SELECT und WHERE Klauseln wie bei SQL. EPL Statements definieren einen oder mehrere Ansichten (views). Innerhalb von diesen Views können Daten abgefragt, gefiltert o.ä. werden, ähnlich wie bei SQL Abfragen.

Ein einfaches Beispiel soll die Funktionsweise verdeutlichen: Ein Aktienhändler möchte Aktien nur dann kaufen, wenn deren Preis unterhalb eines bestimmten Wertes liegt. Er möchte dabei nicht jede einzelne Kursveränderung überwachen, sondern ihn interessiert, die Veränderung innerhalb eines bestimmten Zeitraums, bspw. 30 Sekunden. Eine EPL Abfrage, welche den Durchschnittswert einer Aktie in den letzten 30 Sekunden kalkuliert könnte so aussehen:

```
Select avg(price) from StockTick.win:time(30 sec)  
where symbol = 'GOOG'
```

Das nächste Beispiel zeigt, wie zwei Eventströme miteinander kombiniert werden können. Der erste Eventstrom enthält erkannte Fälschungen in den letzten 30 Minuten. Der zweite Strom enthält Auszüge von den letzten 30 Sekunden. Die Ströme werden kombiniert indem die gleiche Kontonummer herangezogen wird. Damit sollen falsche Abbuchungen von einer bestimmten Kontonummer innerhalb einer bestimmten Zeit erkannt werden. Somit könnte die Bank die entsprechende Kontonummer sperren:

```
select fraud.accountNumber as acctNum, fraud.warning as warn, withdraw.amount as  
amount,  
MAX(fraud.timestamp, withdraw.tmiestamp) as timestamp, , 'withdraw\Fraud' as desc  
from FraudWarningEvent.win:time(30 min) as fraud,  
WithdrawalEvent.win:time(30 sec) as withdraw  
where fraud.accountNumber = withdraw.accountNumber
```

EPL Module

Esper beinhaltet auch EPL Module, eine spezielle Infrastruktur um Eventgetriebene Applikationen zu bündeln und zu entwickeln. Eine EPL Modul Datei ist eine Textdatei. Diese Datei enthält EPL Statements, getrennt durch ein Semikolon (;).

Überblick über die Teile einer EPL Modul Datei:

```
[module module_name;]
[uses module_name; | import import_name;] [uses module_name; | import import_name;]
[...]
[epl_statement;] [epl_statement;] [...]
```

So könnte eine Beispiel EPL Modul Datei aussehen:

```
//Declare the name for the module
module org.myorganization.switchmonitor;

//Declare other module(s) that this module depends on
uses orgmyorgnization.common;

//Import any Java/.NET classes in an application package
import org.myorganization.events.*;

//Declare an event type based on a type in the namespace that was imported as
//above
create schema MySwitchEvent as MySwitchEventObject;

//Sample statement #1
@Name('All-MySwitchEvent')
@Description('Detect each MySwitchEvent')
select * from MySwitchEvent;

//Sample statement #2
@Name('Count-Switched-On')
@Description('Count per switch id of the number of Off-to-On switches in the last
1hour')
select ID, count(*) from MyOffOnStream.win:time(1 hour)
group by ID;
```

5.1.6 Zusammenfassung

CEP kann verwendet werden wenn folgende Kriterien auf die Applikation zutreffen (Chandy, et al., 2011):

Hohe Eingangsrate

Schnelle Evaluierung wird benötigt

Hohe Relevanz von neuen (jungen) Daten

Zeitlicher Zusammenhang von Events ist notwendig

Rohdaten müssten nicht archiviert werden (optional in einigen CEP-Engines)

Beispielapplikationen sind der Umgang mit Businessprozessen, Verkehrsüberwachung, Katastrophen-Frühwarnsystem, finanzielle Transaktionen, autonome Fahrzeuge (Hinze, et al., 2009).

Im Gegensatz zu Datenbanksprachen wie SQL können zeitliche oder kausale Kontextinformationen zwischen den gespeicherten Daten ermittelt werden. Bevor Daten in einem traditionellen Database Management System (DBMS) verarbeitet werden können, müssen diese zunächst (dauerhaft) gespeichert und indiziert werden. Die

Daten werden nur verarbeitet wenn der Benutzer explizit danach fragt. Deshalb können DBMS schlecht die Anforderungen an Applikationen erfüllen welche kontinuierliche und zeitliche Verarbeitung von Information in Echtzeit verlangen (Cugola, et al., 2012).

Message-basierte Systeme wie JMS sind nicht dafür entwickelt worden um mit zeitlich abhängigen Daten und Echtzeit Anfragen umgehen zu können. JMS Systeme sind zustandslos und die Entwickler müssen die Logik selbständig implementieren.

Ein intNAV erfüllt alle genannten Kriterien. Eine hohe Eingangsrate ist gegeben wenn das CEP System auf einem Server läuft wo alle Events von verschiedenen tausenden Benutzern verarbeitet werden. Eine schnelle Evaluierung der Daten ist notwendig um in der Lage zu sein den Benutzer in Echtzeit über Probleme auf seiner Route zu informieren. Für das intNAV werden ausschließlich neue (junge) Daten benötigt, da sonst keine zuverlässige Echtzeit Navigation möglich ist. Die zeitliche Korrelation von Events kann benutzt werden um beispielsweise die durchschnittliche Geschwindigkeit basierend auf GPS-Einträgen zu berechnen. Archivierung von Rohdaten ist nicht notwendig, kann jedoch nützlich sein um beispielsweise alle GPS-Einträge auf einer Karte zu zeigen. Aus diesem Grund wurde CEP für die Modellierung eines intNAVs herangezogen.

5.2 KONZEPT

Basierend auf dem im vorhergehenden Kapitel definierten theoretischen Grundlagen, soll nun das Konzept für das intNAV vorgestellt werden. Im letzten Unterkapitel 5.1.6 wurde festgestellt, dass CEP fast alle Kriterien erfüllt um als EDA für das intNAV Genüge zu tun. Ein intNAV soll folgende Anforderungen erfüllen:

Event Processing Language (EPL) Aussagen sollen in EPL Module ausgelagert werden.

Nur relevante Services sollen gestartet werden um die Central Processing Unit (**CPU**) sowie den Internetverbrauch nicht zu überlasten.

Die Anwendung soll bei festgestellten Abweichungen (passiv und aktiv) passende Alternativen für den Benutzer liefern.

In **Abbildung 45** wird die Architektur basierend auf den Anforderungen an ein intNAV unter Verwendung von EDA dargestellt. Die konzeptionelle Architektur besteht aus der Hauptapplikation, einem CEP System und EPL Modulen. Wenn die Hauptapplikation zum ersten Mal ausgeführt wird, besteht der erste Schritt darin, die Route mittels des Services Google Directions Application Programming Interface²⁰ (**API**) zu planen. Basierend auf den Transportmitteln auf der geplanten Route, werden die entsprechenden EPL Module generiert. Wenn eine Route beispielsweise die Modalitäten „Gehen“ und „ÖPNV“ enthält, dann werden die EPL Module „All transport modes“, „Walking“ und „Transit“ generiert, jedoch nicht das „Driving“ Modul, da die geplante Route keinen

²⁰ <https://developers.google.com/maps/documentation/directions/?csw=1>

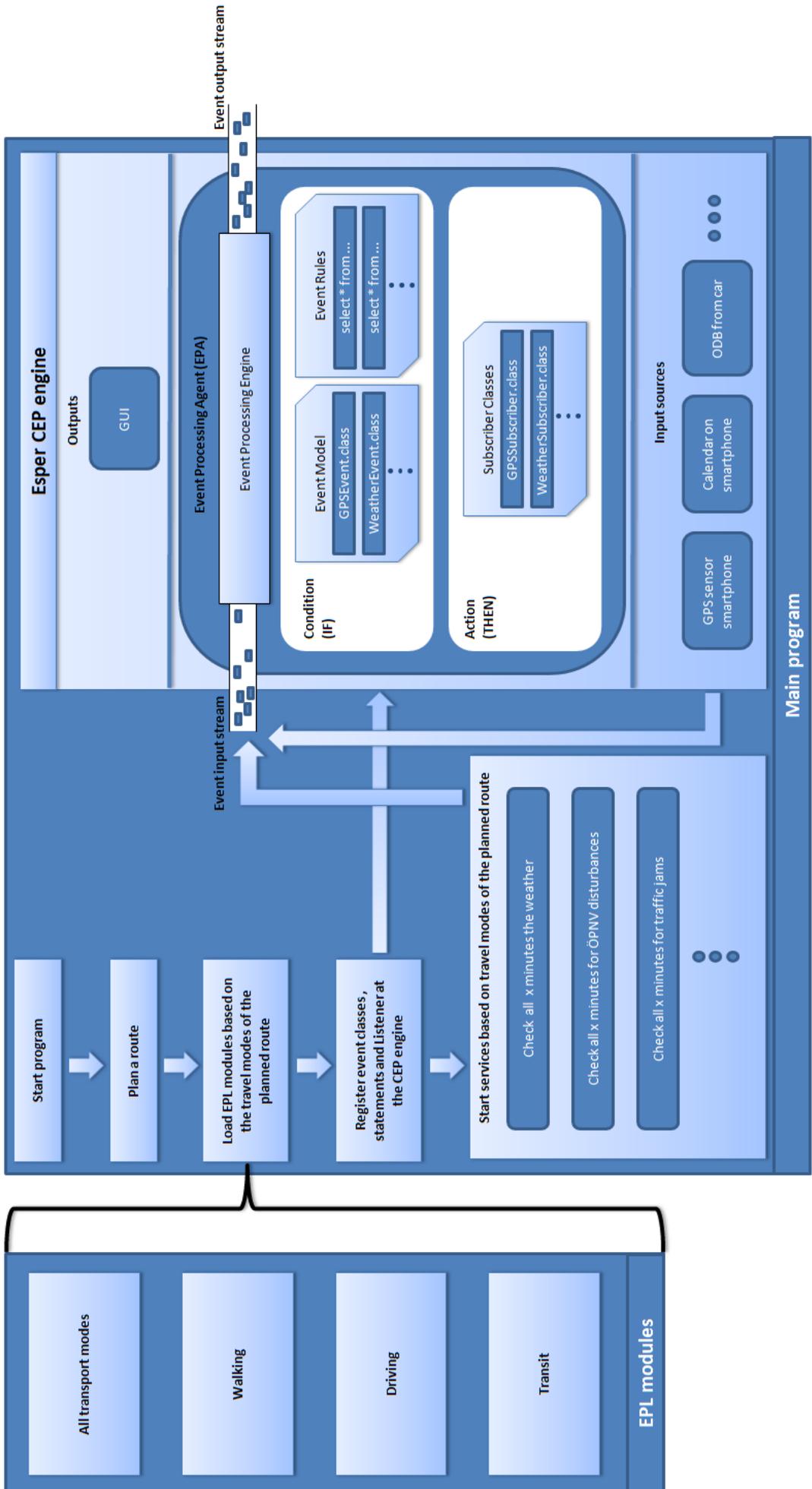


Abbildung 45: CEP-Architektur eines intNAVs

Abschnitt enthält, der mit dem Auto befahrbar wäre. Basierend auf den generierten Modulen werden die Eventklassen, die Statements und die „Subscriber“ Klassen auf dem CEP System registriert. Zusätzlich werden alle relevanten Adapter Klassen gestartet, welche Events generieren die zum Event-Eingangsstrom des CEP Systems gesendet werden. Neben diesen Adapterklassen gibt es noch weitere Eingangsgrößen für den Eventeingangsstrom, wie z.B. die Aufzeichnungen des GPS-Sensors. Mit diesem Ansatz werden die Anforderungen 1 und 2 erfüllt. Das einzige Resultat des CEP Systems ist in diesem Falle die GUI der Applikation in welcher dem Benutzer Empfehlungen und weitere Informationen gezeigt werden.

Das CEP System besteht aus einem Eventverarbeitungsagenten (EPA) welcher in Kapitel 5.1.3 beschrieben wurde. Dieser Agent beinhaltet Event Module welche in Paketen für jedes Transportmodul definiert sind (ähnlich wie die EPL Modulen „All transport modes“, „Walking“, „Driving“ und „Transit“). Diese Pakete werden in die EPL Modul Dateien geladen welche die entsprechenden Eventregeln beinhalten. Für jedes Statement der ausgeführten EPL Module wird eine Subscriber Klasse kreiert welche eine „Update Listener Methode“ enthält. Diese Methode führt den Ausführungsteil (then) des EPL Statements aus, wenn der Bedingungsteil (if) im Eventstrom erkannt wurde.

EPL Module wurden im Kapitel 5.1.5 genauer beleuchtet. Diese werden in der hier vorgestellten Architektur dafür verwendet um EPL Statements, abhängig von den verfügbaren Transportmitteln auf der geplanten Route, zu laden. Das EPL Modul „All transport modes“ wird dabei immer ausgeführt. Es enthält EPL Statements welche für jede Route gültig sind. Die Module „Walking“, „Driving“ und „Transit“ enthalten spezielle Statements für den entsprechenden Transportmodus.

Folgende Adapterdienste sind in der konzeptuellen Architektur verfügbar, welche die Anforderung 3 erfüllen:

DeviationProbabilityService („Walking“, „Driving“): berechnet die Abweichungswahrscheinlichkeit.

LocationService („Walking“, „Driving“): holt sich in regelmäßigen Abständen die GPS Lokation.

MVVCrawlerService („Transit“): überprüft die Abfahrtszeiten der Münchner Verkehrsgesellschaft (MVG) in 10, 5 und 1 Minute vor dem Starten des nächsten Routenabschnitts.

OEPNVPredictionService („Transit“): erkennt öffentliche Verkehrsmittel auf der Oberfläche (S-Bahn, Bus und Tram).

TimeBeforeRouteStartService („All transport modes“): meldet dem Benutzer wenn dieser anfangen sollte zum Startpunkt loszugehen.

UserAtEndLocation („Walking“, „Driving“): untersucht, ob der Benutzer das Ziel seiner Route in der geplanten Zeit erreichen kann.

WeatherService („Walking“): überprüft ob es auf den Abschnitten, die der Benutzer zu Fuß zurücklegen muss, regnet.

5.3 IMPLEMENTIERUNG

Dieses Unterkapitel soll zeigen, wie das vorgestellte Konzept im vorhergehenden Unterkapitel 5.2 implementiert wurde. In Unterkapitel 5.3.1 wird das Google Directions Application Programming Interface (API) beschrieben, welches verwendet wurde, um die intermodale Route zu planen. Die Implementierung von Esper wird in Unterkapitel 5.3.2 beschrieben und in Kapitel 5.3.3 werden die zwei entwickelten Prototypen detailliert beschrieben.

5.3.1 Google Directions API

Um eine Route mit Hilfe des intNAVs planen zu können, wird eine Routenplanungssoftware benötigt. Die Google Directions API²¹ kann dafür verwendet werden Routen auf fast die gleiche Weise zu planen, wie es mit der Google Maps Applikation möglich ist. Diese kann kostenlos mit einigen Restriktionen verwendet werden (2500 Routenanfragen pro Tag, 2 Anfragen pro Sekunden und bis zu 8 Wegepunkte für jede Anfrage). Mittels dem Hypertext Transfer Protocol (**HTTP**) kann die Google Directions API angefragt werden:

```
https://maps.googleapis.com/maps/api/directions/<output><?parameters>
```

Der <output> Parameter kann entweder `json` für JavaScript Object Notation (**JSON**) oder `xml` für Extensible Markup Language (**XML**) sein. Es gibt einige optionale und nicht optionale Parameter <?parameters> welche hinzugefügt und mittels (&) getrennt werden können. Die folgende Liste enthält die entsprechenden Parameter:

`origin`: Startadresse als String oder mittels Latitude/Longitude Koordinaten.

`destination`: Zieladresse als String oder mittels Latitude/Longitude Koordinaten

`sensor`: dieser Parameter definiert, ob die Abfrage von einem Device mit einem Location-Sensor ist. Der Parameter kann `true` oder `false` sein.

Eine weitere Liste der optionalen Parameter:

`mode`: definiert den Transportmodus, z.b. `walking`, `driving` oder `transit`

`alternatives`: es werden mehr als nur eine route geladen, wenn dieser Parameter auf `true` gesetzt wurde.

`departure_time`: die gewünschte Startzeit als integer in Sekunden Start Mitternacht. 1. Januar 1970 Coordinated Universal Time (**UTC**)

`arrival_time`: die gewünschte Ankunftszeit als integer in Sekunden Start Mitternacht. 1. Januar 1970 Coordinated Universal Time (**UTC**)

`transit_mode`: spezifiziert eine oder mehr bevorzugte Transportmodi. Beispielsweise Bus, Tram oder S-Bahn.

`transit_routing_preferences`: spezifiziert bevorzugte Routen, zwei mögliche Parameter sind: `less_walking`, `fewer_transfers`.

Um nun diese zwei letzten Parameters verwenden zu können, wird ein zusätzlicher **API** Schlüssel zu der http-Abfrage hinzugefügt. Eine mögliche Abfrage könnte demnach so aussehen:

```
https://maps.googleapis.com/maps/api/directions/json?origin=Hanauer+Strasse+46,+80992+Munich&destination=Pasing,+Germany&departure_time=1427356560&mode=transit&alte
```

²¹ <https://developers.google.com/maps/documentations/directions/>

rnatives=true&transit_mode=&key=BlzaSyCmSgTioE_GZJySwEB9F0fhrYwoLLCT2es&sensor=true

Ein Beispielresultat enthält ein Array aus „legs“. Eine Route besteht aus einem „leg“, wenn es keine Wegepunkte gibt. Jeder „leg“ enthält ein Array von „steps“ und weitere Information. Ein „step“ ist der kleinste Bestandteil (atomar) einer Route und enthält spezifische Instruktionen. **Abbildung 46** zeigt eine graphische Repräsentation einer Route mit einem „leg“ und mehreren „steps“.

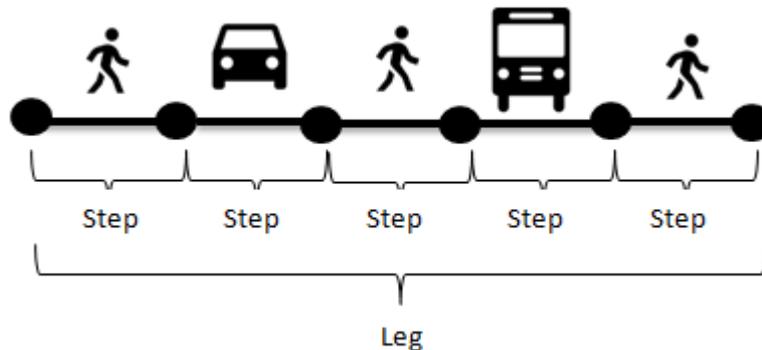


Abbildung 46: Eine Route aus Google Directions API aufgeteilt in „Legs“ und „Steps“.

5.3.2 Esper

Um eine CEP Applikation mittels Esper designen zu können, bietet die offizielle Webseite²² eine Schritt-für-Schritt Anleitung:

Definition des Ziels der Anwendung.

Definition der Situationen welche untersucht werden oder welche Information berichtet werden soll.

Definition der Performanz Anforderungen, insbesondere Durchsatz und Latenz.

Definition des Ursprungs der Events.

Definition des Formats von einfachen Events.

Definition von Eventregeln.

Definition, in welcher Form Events vorliegen: Java, Maps oder **XML**.

Definition von EPL Statements um Muster zu erkennen.

Verwendung von Comma-Separated Values (**CSV**) als Event Simulator, um die Situationen welche untersucht werden zu testen.

Es soll gegen die Performanz Anforderungen getestet werden: Durchsatz und Latenz in der Zielumgebung.

Die nachfolgenden Sektionen basierend auf der Liste von Esper. Dort wird beschrieben wie das Applikationskonzept implementiert wurde.

5.3.2.1 ANFORDERUNGSANALYSE

²² <http://www.espertech.com/esper/tutorial.php>

CEP wird nicht auf einem Server sondern auf einem Client verwendet, beispielsweise auf einem Smartphone. Heutzutage sind die meisten Smartphones mit einer sehr guten Hardware ausgestattet, können jedoch nicht mit der Rechenleistung von Computern konkurrieren. Das größte Problem ist der Akku, da die hohe Rechenleistung zu einem hohen Akkuverbrauch führt. Smartphones bieten die Möglichkeit, sich mit dem Internet zu verbinden aber die Verbindung ist nicht sehr zuverlässig. Diese Punkte sollen beachtet werden, wenn eine Applikation auf einem mobilen Endgerät entwickelt werden soll. Ein Smartphone besitzt einen integrierten **GPS** Sensor, welcher es erlaubt die aktuelle Position u ermitteln – dies ist sehr wichtig für ein intNAV (Stipkovic, 2012).

Ein Review des Fraunhofer Instituts zeigte, dass es kein offizielles Event-Verarbeitungssystem für mobile Endgeräte gibt (Vidackovic, et al., 2010), da die Hersteller davon ausgegangen sind, dass die Applikationen mit CEP auf großen Servern entwickelt werden. Jedoch gibt es einige Entwickler welche Schnittstellen für mobile Systems entwickelt haben. Für Android existiert genau eine solche Software, Asper²³ basierend auf Esper.

CEP auf mobilen Endgeräten kann dabei auf zwei Weisen verwendet werden (**Abbildung 47**):

CEP auf dem Smartphone

CEP auf einem externen Server.

Wenn jedoch, wie für das intNAV benötigt, Echtzeitdaten benötigt werden, kann es passieren, dass aufgrund von Verbindungsproblemen zu externen Servern, die Berechnungen nicht mit den neuesten Daten gemacht werden.

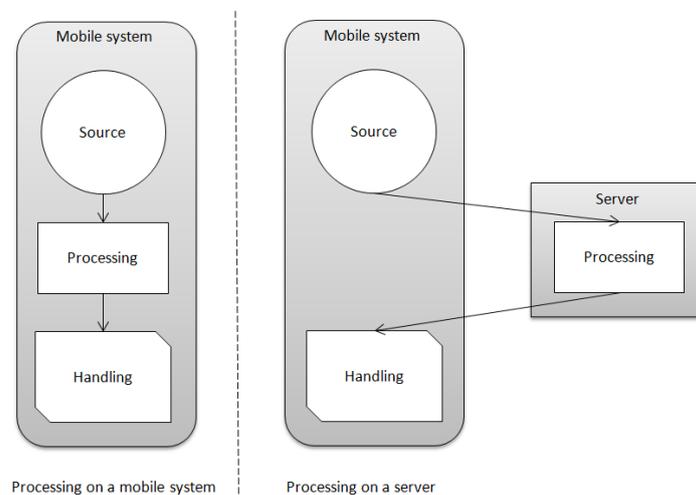


Abbildung 47: CEP auf einem mobilen System und auf einem Server. Wenn die CEP auf einen Server ausgelagert wird, ist die Verbindung nicht zuverlässig und Daten können verloren gehen (nach (Stipkovic, 2012)).

Ziel der Anwendung

²³ <https://github.com/mobile-event-processing/Asper>

Im ersten Schritt sollen die Situationen welche untersucht werden sollen analysiert werden. Das System soll in der Lage sein, diese Situationen zu erkennen und darauf angemessen zu reagieren. Die möglichen Situationen, sowie sie aktiven und passiven Abweichungen wurden bereits ausgiebig in Kapitel 4 behandelt. Ausschnittsweise sollen hier ein paar genannt werden:

Der Benutzer geht nicht zur geplanten Startzeit los

Der Benutzer steigt in das falsche Transportmittel ein

Aufgrund von Verspätungen im **ÖPNV** kann die Route nicht wie geplant benutzt werden

Den Benutzer können nun folgende Fragen interessieren:

Kommt der ÖPNV pünktlich?

Erreiche ich die Abfahrtszeit pünktlich?

Bin ich in das korrekte Transportmittel gestiegen

Bin ich noch gut in der Zeit oder muss ich mich beeilen, um zu meinem ÖPNV zu gelangen?

Basierend auf diesen Anforderungen kann eine Liste von möglichen Events definiert werden, welche in **Tabelle 11** gelistet sind.

Transportart	Event	Beschreibung
Alle	CurrentGPSLocationEvent	Empfange GPS Daten
Alle	CurrentGPSLocationEvent	Berechne durchschnittliche Geschwindigkeit von den aufgezeichneten GPS Daten
Alle	DeviatedFromRouteEvent	Untersuche, ob der Benutzer von der geplanten Route abgewichen ist.
Alle	CurrentStepEvent	Untersuche, ob der Benutzer die Startposition eines „Steps“ auf der geplanten Route erreicht hat.
Alle	BadWeatherOnRouteEvent	Untersuche, ob schlechtes Wetter auf dem Abschnitt „Fußweg“ der geplanten Route ist.
Alle	NotYetMovedoffStartLocationEvent	Untersuche, ob sich der Benutzer frühzeitig an der Startposition der geplanten Route befindet.
Zu Fuß	CurrentWalkingStepEvent	Untersuche, ob sich der Benutzer an der Startposition des Abschnittes „Fußweg“ befindet.

Zu Fuß	CurrentWalkingStepEvent und OEPNVDisturbanceOnRouteEvent	Überprüfe, ob es zu einer Störung im ÖPNV kommt, nach dem Abschnitt „Fußweg“
Zu Fuß	CurrentWalkingStepEvent und TrafficJamOnRouteEvent	Überprüfe, ob es einen Stau auf dem Abschnitt „Fahren“ nach dem Abschnitt „Fußweg“ gibt.
Fahren	CurrentDrivingStepEvent	Untersuche, ob sich der Benutzer an der Startposition des Abschnittes „Fahren“ befindet.
Fahren	TrafficJamOnRouteEvent	Überprüfe, ob es einen Stau auf der geplanten Route gibt.
Fahren	CurrentDrivingStepEvent und OEPNVDisturbanceOnRouteEvent	Überprüfe, ob es eine Störung im ÖPNV gibt, nach dem Abschnitt „Fahren“.
ÖPNV	CurrentTransitStepEvent	Untersuche, ob sich der Benutzer an der Startposition des Abschnittes „Transit“ befindet.
ÖPNV	OEPNVDisturbanceOnRouteEvent	Überprüfe, ob es eine Störung im ÖPNV auf der geplanten Route gibt.
ÖPNV	CurrentOEPNVTransportationEvent	Untersuche, ob der Benutzer ein Verkehrsmittel des ÖPNV verwendet.

Tabelle 11: Überblick über die verwendeten Events im Prototypen

Performanz

Die Anwendung welche hier entwickelt werden soll, soll in der Lage sein komplexe Events in Echtzeit zu erkennen. Wenn beispielsweise der geplanten Route ein Segment enthält auf dem Stau vorherrscht und der Benutzer sich am Beginn dieses Segments befindet, sollte dieser sofort informiert werden, sodass der Benutzer den verursachten Zeitverlust aufholen kann. Ein weiteres Beispiel, ist ein Segment auf welchem der Benutzer zu Fuß unterwegs ist, bei dem jedoch Regen festgestellt wird. Das System sollte den Benutzer darüber so früh wie möglich in Kenntnis setzen und ihm Alternativen anbieten, welche keine langen Fußmärsche beinhalten.

Esper kann mit bis zu 500.000 Events/s auf einem 2GHz Prozessor (CPU) umgehen (Vasseur, 2014). Folgende Raten werden bei einem Benutzer erwartet:

GPS: 1 Event/s

Abweichung: 1 Event/s

Münchner Verkehrsgesellschaft (MVG) Abweichung: 1 Event/s

Schritt: 1 Event/s

In Summe darf also von maximal 10 Events/Sekunde pro Benutzer ausgegangen werden, sodass der maximale Durchsatz von 500.000 Events/Sekunde von Esper resp. Asper bei weitem nicht erreicht wird. Wenn jedoch dieser CEP Ansatz auf einem externen Server mit mehreren Clients laufen würde, müsste der maximale Durchsatz analysiert werden.

Ursprung des Events

Die Events für diese Applikation stammen aus verschiedenen Quellen:

GPS Position wird über den GPS Sensor des Smartphones bezogen.

Störungen im ÖPNV werden über die MVG-Live Webseite²⁴ bezogen.

Erkennung des Transportmittels wird über den Algorithmus aus Kapitel 3 bezogen.

Abweichungswahrscheinlichkeit wird über den Algorithmus aus Kapitel 4 bezogen.

5. Definition des Formats von einfachen Events

Einfache Events wurden in Kapitel 5.1.3 beschrieben. Die verwendeten Events in dieser Anwendung sind in dem UML-Diagramm in **Abbildung 48** dargestellt. Alle Eventklassen erweitern die `AbstractEvent` Klasse, welche die wichtigen Attribute eines Events beinhaltet: eine eindeutige ID (ID), und einen Zeitstempel. Die Bedeutung eines jeden Events ist in **Tabelle 11** dargestellt.

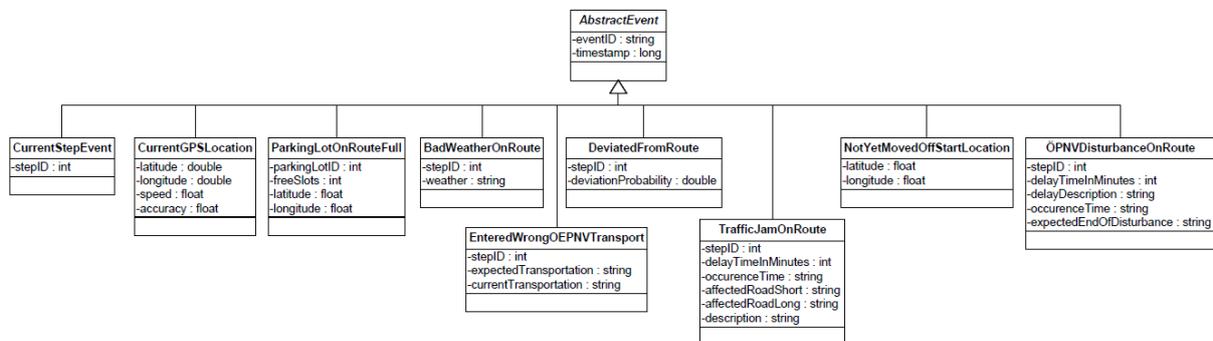


Abbildung 48: UML Diagramm zeigt die Events, welche in dem Prototypen verwendet wurden.

6. Format von einfachen Events

Ein Vergleich über die verfügbaren Esper Event Repräsentationen ist in **Tabelle 12** dargestellt. Die relevanten Parameter sind in dieser Sektion beschrieben.

Events werden als Plain Old Java Objects (**POJOs**) dargestellt weil sich diese durch gute Performanz auszeichnen und weniger Speicher benötigen. Der wichtigste Grund ist jedoch die Performanz. Lewis und Kane von der Yale Universität berichteten auf der Finance 2010 Konferenz über einen Durchsatz von ca. 4.000 Events/Sekunde mittels XML-beschriebenen Events²⁵. Über 250.000 Events/Sekunden konnten auf der gleichen Hardware mit POJO Events erreicht werden.

²⁴ <http://mvg-live.de>

²⁵ <http://illposed.net/esperr.html>

Um den Speicherverbrauch eines Java Objects zu beispielsweise einer HashMap zu vergleichen, wird folgendes Beispiel herangezogen:

```
class X{
int a;
byte b;
}
```

8 Bytes werden für die Referenz der Klassendefinition verbraucht, 1 Byte für die Variable b, 4 Bytes für die Variable a und 3 zusätzliche Bytes für das Padding um auf 16 aufzurunden, damit durch 8 geteilt werden kann. Somit verbraucht diese Java Object 16 Bytes des Speichers.

Eine HashMap beinhaltet einen Schlüssel, Wert, Hash eines Schlüssels, einen int und einen Zeiger auf den nächsten Eintrag. Ein Eintrag nimmt somit 32 Bytes in Anspruch: 12 Bytes für den Header, 16 Bytes für die Daten und 4 Bytes für das Padding. Eine HashMap der Größe = N benötigt also 32*N Bytes Speicherplatz.

	Java Object (POJO/Bean oder andere)	Map	Object-Array	XML-Dokument
Performanz	Gut	Gut	Sehr gut	Nicht vergleichbar und hängt von dem verwenden von XPath ab
Speicherverbrauch	Wenig	Medium	Wenig	Hängt von DOM und XPath Implementierung ab, kann viel sein.
Insert-into that Repräsentation	Ja	Ja	Ja	Nein
Create-schema Syntax	Ja	Ja	Ja	Nein

Tabelle 12: Vergleich möglicher Eventrepräsentationen in Esper ((Inc., 2017), Seite 36)

5.3.3 Prototyp

Im Rahmen der vorliegenden Thesis wurden 2 Prototypen ausgearbeitet um sich mit dem Esper Framework anzufreunden und die Funktionalität von CEP für intNAVs zu testen.

5.3.3.1 JAVA PROTOTYP

Ein erster Prototyp wurde in Java entwickelt um sich mit dem Esper Framework bekannt zu machen und um zu sehen, ob die EDA für intNAVs verwendet werden kann. Der Prototyp und die implementierte CEP wurde auf einem stationären Computer und nicht auf einem mobilen Endgerät entwickelt. Für diesen Prototyp wurden nur simulierte Daten verwendet. Das Programm zeigt zwei statische Routen mit unterschiedlichen Transportoptionen und enthält keine Möglichkeit um intermodale Routen zu planen. Wenn das Programm

gestartet wird, erscheinen Punkte für jedes erkannte Event. Das Resultat ist eine Empfehlung in dem Textfeld. **Abbildung 49** zeigt einen Screenshot des Java Prototypen.

Die Test Applikation wurde entwickelt um zu testen ob komplexe Events in dem Kontext von intNAVs erkannt werden und die entsprechende Output Empfehlung zum User gesendet werden kann. Das Beispiel in **Abbildung 49** zeigt eine Empfehlung basierend auf dem erkannten Regen auf dem Abschnitt auf dem der Benutzer zu Fuß gehen würde. Die Empfehlung wurde gesendet, als der simulierte Benutzer im Auto gefahren ist und die Tankfüllung ausreichend war um zum Ziel zu fahren. Zusätzlich wurden die restlichen Routenabschnitte dahingehend überprüft ob auf ihnen Staus vorliegen und ob der geplante Termin rechtzeitig erreicht werden kann.

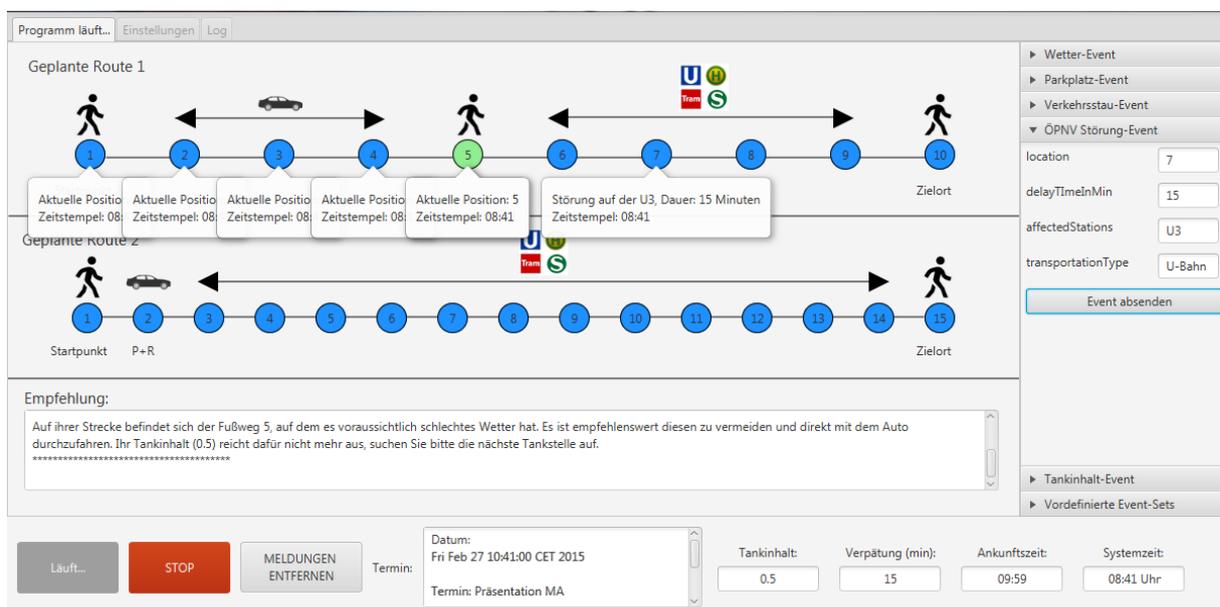


Abbildung 49: Screenshot des Java Prototypen welcher auf einem lokalen Computer und nicht auf einem mobilen Endgerät entwickelt wurde. Dabei wurden nur simulierte Daten verwendet. Der Prototyp wurde entwickelt um zu testen ob komplexe Events in intNAVs entdeckt werden können und die entsprechenden Empfehlungen ausgegeben werden.

Mit Hilfe dieser Simulation mittels CEP ist der Benutzer in der Lage anstelle der ursprünglichen Route, bei welcher er höchstwahrscheinlich nass geworden wäre, doch noch trocken und pünktlich zum Termin zu kommen.

Da diese Anwendung ausschließlich dazu diente, die Eignung von CEP für intNAVs zu testen, wurden auch keine Benchmark Tests durchgeführt um den Durchsatz und die Latenz von Esper CEP in diesem Kontext zu überprüfen. Dieser Prototyp hat bestätigt, dass CEP für die hier vorliegende Arbeit angewendet werden kann.

5.3.3.2 ANDROID PROTOTYP

Ein voll-funktionsfähiger Android Prototyp wurde im Rahmen der vorliegenden Thesis entwickelt um die Funktionsweise eines intNAVs sowie die Anwendbarkeit von CEP für intNAVs auf mobilen Endgeräten zu testen. Android wurde deshalb als mobiles Betriebssystem ausgewählt, da es sehr beliebt ist und in Deutschland einen

Marktanteil von 70,1%²⁶ inne hält. Die Entwicklung für Android ist kostenfrei und verfügt über eine große Anzahl an Open-Source Bibliotheken und Frameworks.

Wenn die Anwendung zum ersten Mal ausgeführt wird, erscheint ein Informationsdialog (vgl. **Abbildung 50**) um danach einen Startbildschirm zu zeigen, während die aktuelle Position über GPS bezogen wird (solange GPS eingeschaltet ist) und die tempv16.db Datei aus dem assets Ordner in dem Download Ordner auf der SD Karte des Smartphones kopiert ist. **Abbildung 51** zeigt den Startbildschirm. Die Datenbankdatei (tempv16.db) wird benötigt, damit die Transportmittelerkennung funktioniert. Sie enthält zusätzlich die komplette Infrastruktur des öffentlichen Verkehrsnetzes an der Oberfläche von München. Nach dem Startbildschirm wird die Aktivität PlanRouteActivity ausgeführt, in welcher der Benutzer eine Route mittels Google Directions API planen kann.

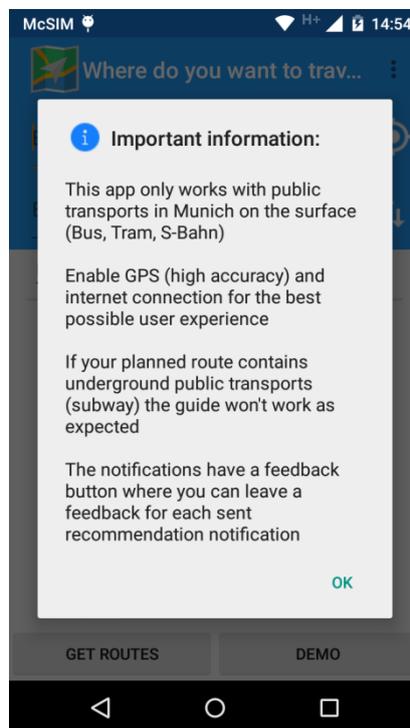


Abbildung 50: Wenn die Applikation zum ersten Mal ausgeführt wird, erscheint folgender Informationsdialog.

²⁶ <http://www.kantarworldpanel.com/global/smartphone-os-market-share>



Abbildung 51: SplashScreenActivity: diese Aktivität wird aufgerufen, wenn die Applikation gestartet wird. Wenn GPS bei Applikationsstart aktiviert ist, wird die aktuelle Position in der Datenbank abgespeichert.

Wenn der Benutzer den Get Routes Button betätigt, erscheint eine Liste von möglichen Routen. Diese werden von der Google Directions API bezogen.

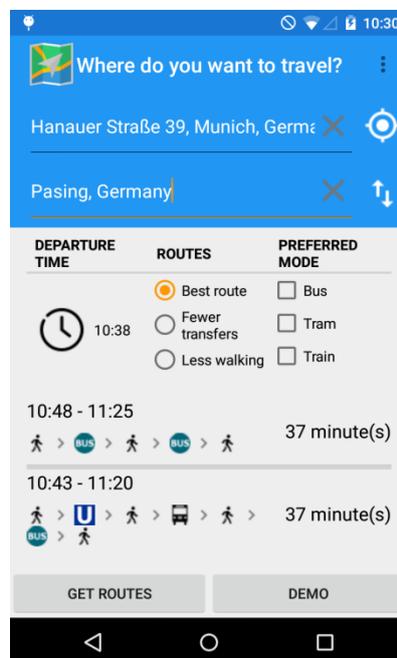


Abbildung 52: PlanRouteActivity: diese Aktivität wird benötigt um eine Route mittels Google Directions API zu planen.

Wenn eine Route ausgewählt wird, wird die Aktivität RouteGuideActivity gezeigt (vgl. **Abbildung 53**). Der Routenführer wird automatisch gestartet, somit auch alle relevanten Hintergrundservices. Auf der linken Seite des RouteGuideActivity Displays wird die Route in einer scrollbaren Ansicht dargestellt. Wenn der Benutzer sich an der Startposition eines „steps“ auf der Route befindet, wird der aktuelle „step“ grün markiert. Die rechte Seite des Bildschirms beinhaltet mehrere Textfelder in einer scrollbaren Ansicht um den Benutzer über bestimmte Events, bspw. die aktuelle Position, eine Empfehlung, das aktuelle Transportmittel und die Abweichungswahrscheinlichkeit, zu informieren. Im unteren rechten Teil des Bildschirms ist ein zusätzlicher Container, in welchem manuell Events an die CEP geschickt werden können. Dies ist hauptsächlich aufgrund von Debugging und Testzwecke implementiert worden. Im unteren Teil des Bildschirms werden mehrere Alternativrouten angezeigt. Die erste Alternative ist die zweitbeste Alternative zu der Routenoption „Best Route“, die zweite Alternative ist die erste Route von der Option „weniger Transfers“ und die dritte Alternativroute ist die beste Route von der Option „weniger gehen“. Um beispielsweise Regen auf der Strecke, auf der der Benutzer zu Fuß unterwegs ist, zu visualisieren, wird ein Warndreieck im entsprechenden „step“ gezeigt (vgl. **Abbildung 54**).



Abbildung 53: RouteGuideActivity: diese Aktivität wird während der gesamten intermodalen Navigation gezeigt. Auf der linken Seite sieht man die geplante Route und der markierte grüne Abschnitt zeigt an, in welchem Abschnitt der intermodalen Route sich der Benutzer befindet.

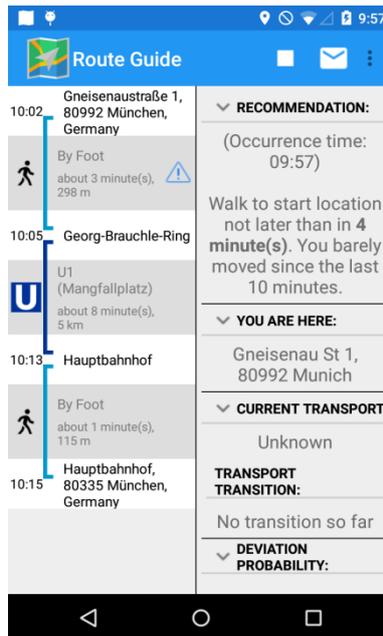


Abbildung 54: RouteGuideActivity: Ein Warndreieck im ersten Abschnitt informiert den Benutzer über ein Problem auf diesem Abschnitt. Wenn das Warndreieck angeklickt wird, wird die entsprechende Information angezeigt.

Wenn der Text innerhalb des Empfehlungstext-Feldes aktualisiert wird, erscheint eine Benachrichtigung, wie in **Abbildung 55** zu sehen ist.

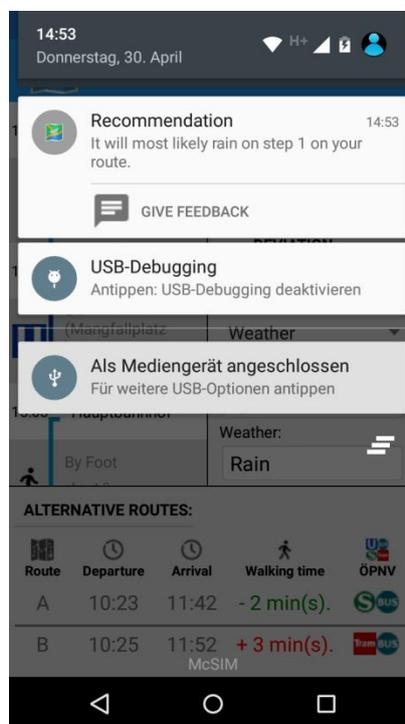


Abbildung 55: Für jede Empfehlung wird eine Android Benachrichtigung generiert.

Der obere Teil der Anwendung (**Abbildung 53** und **Abbildung 54**) enthält einen Knopf um den Routenführer zu stoppen, einen weiteren Button um die Aktivität `FinalFeedbackActivity` zu zeigen. Die `FinalFeedbackActivity` wird verwendet um Feedback von dem Benutzer über die GUI zu erhalten (vgl. **Abbildung 56**). In Kapitel 5.4 wird genauer darauf eingegangen. **Abbildung 57** zeigt die Aktivität `SettingsActivity` in welchen der Benutzer die Applikationseinstellungen verändern kann, z.B. soll das Smartphone bei jeder Startposition eines „steps“ vibrieren. Zusätzlich kann die Datenbank als CSV Datei exportiert oder gelöscht werden.

Die wichtigsten Klassen des Routenführers sind die Android Services welche zusammen mit der Aktivität `RouteGuideActivity` gestartet werden. Diese wurden in Kapitel 5.2 genauer beschrieben. Der Service `DeviationProbabilityService` wird innerhalb der Java Klasse `CurrentWalkingStepEventSubscriber` gestartet. Die Wahrscheinlichkeit für eine Abweichung wird in jedem „step“ der geplanten Route berechnet. Die Wahrscheinlichkeit für eine Abweichung wird an die GUI von der Aktivität `RouteGuideActivity` gesendet. Damit eine vernünftige Abweichungswahrscheinlichkeit berechnet werden kann, wurde die GPS Abtastrate auf 5 Sekunden aktualisiert, wenn der Benutzer zu Fuß unterwegs ist. Der Wert für die Wahrscheinlichkeit der Abweichung in der GUI wird jede Sekunde aktualisiert. Wenn der Benutzer den „step“ Gehen verlässt, wird das GPS Abtastintervall auf den vom Benutzer festgesetzten Wert zurückgesetzt.

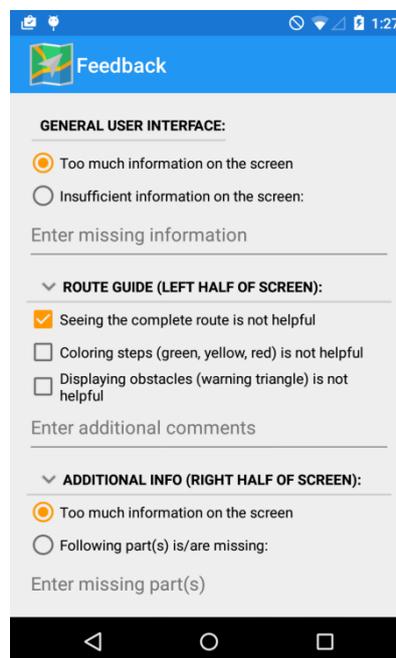


Abbildung 56: `FinalFeedbackActivity`: diese Aktivität wird ausgeführt, um von dem Benutzer ein Feedback über die GUI des Android Prototypen zu erhalten.

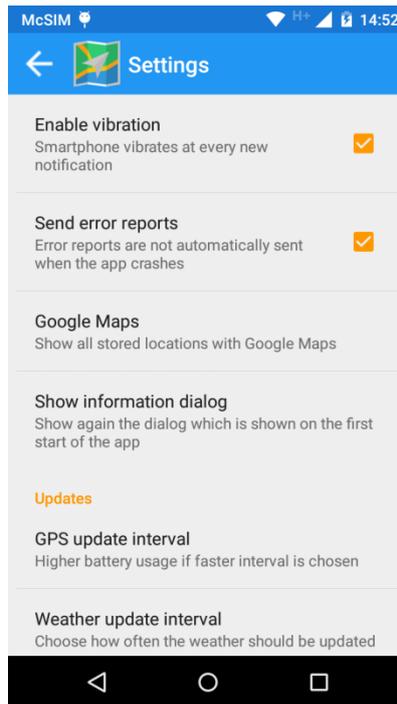


Abbildung 57: SettingsActivity: der Benutzer kann seine Einstellung in der Applikation ändern.

Der Service `LocationService` ist dafür verantwortlich, dass die GPS Positionen vom Smartphone in einem vordefinierten Intervall bezogen werden und die entsprechenden Adressen dazu hinterlegt werden. Die Information wird in der internen Datenbank gespeichert und wird an die GUI durch die Aktivität `RouteGuideActivity` gesendet.

`LocationService` wird gestartet, wenn der nächste Schritt auf der Route den Transportmodus `TRANSIT` beinhaltet. Der Service überprüft alle 15, 10, 5 und 1 Minute bevor dieser Schritt startet ob das ÖPNV Verspätung hat. Die Information über eine mögliche Verspätung wird über die MVG Webseite²⁷ bezogen, wo die Abfahrtszeiten gesammelt werden. Danach wird die Zeitdifferenz zwischen der Abfahrtszeit der geplanten Route sowie der Abfahrtszeit lt. MVG Webseite ermittelt.

Um den aktuellen Transportmittelmodus (oberirdisch) zu erkennen, wird der Service `OEPNVPredictionService` beim Routenführerstart ausgeführt (vgl. Kapitel 3).

`TimeBeforeRoutestartService` startet zusammen mit dem Routenführer und läuft so lange wie der Benutzer noch nicht an der Startposition des ersten Abschnitts der geplanten Route angekommen ist. Dieser Service überwacht die Position des Benutzers bevor der erste Abschnitt startet und schickt eine Benachrichtigung an den Benutzer, damit sich dieser auf den Weg machen soll.

²⁷ <http://www.mvg-live.de>

Um herauszufinden ob der Benutzer an der Zielposition eines Gehabschnitts angekommen ist, wird der Service `UserAtEndLocationService` bei jedem Gehabschnitt ausgeführt.

Das Projekt besteht aus folgenden Paketen:

activities
adapter
cep
events
alltransportmodes
walking
driving
transit
inadapter
outadapter
subscriber
alltransportmodes
walking
driving
transit
database
interfaces
model
services
utils
view

Das **activities** Paket enthält alle Android Aktivitäten welche in der Prototyp Anwendung verwendet werden. Im **adapter** Paket befinden sich Klassen mit deren Hilfe sich scrollbare Objekte darstellen lassen. Das **cep** Paket enthält alle Klassen für das CEP System sowie Klassen um mit weiteren Events umzugehen. Alle Klassen mit Eventdefinitionen befinden sich **events**. Das **inadapter** und **outadapter** Paket beinhaltet die in- und out-Adapter Klassen des CEP Systems. Die Klassen mit den korrespondierenden `updateListener` Methoden befinden sich in dem **subscriber** Paket. Innerhalb des **database** Pakets befinden sich zwei SQLite Datenbanken; das interface Paket enthält die **interface** Klassen. Das **model** Paket besteht aus mehreren **POJO** Klassen welche das Datenmodell definieren. Das **service** Paket enthält alle Serviceklassen. Benutzerlayouts befinden sich in dem **views** Paket und das **utils** Paket beinhaltet einige Helferklassen.

Der **assets** Ordner enthält die Ordner **epl_modules** welche die vier EPL Module `all_vehicle_modules.epl`, `driving_module.epl`, `transit_module.epl` und `walking_module.epl` enthalten.

5.4 BENCHMARKTEST

Der Eventverarbeitungsagent (EPA) innerhalb des Android Prototypen muss mit vergleichsweise wenig Events umgehen. Wie in Abschnitt 5.3.2.1 Anforderungsanalyse beschrieben ist, liegt die maximale Anzahl an Events welche für diesen Prototypen erwartet wurde bei 10 pro Sekunde. Dabei kann das Framework CEP nicht seine volle Stärke ausspielen, da es mit Events von bis zu 500.000 pro Sekunde umgehen kann. Das Ziel des vorliegenden Kapitels war es, die Anwendbarkeit von CEP für intNAVs zu testen. Im zweiten Schritt kann über eine Server-seitige Implementierung des intNAVs diskutiert werden, bei welcher die Daten von mehreren

tausenden Benutzern verarbeiten werden sollen. Da der hier vorgestellte Prototyp nur in München funktioniert, wurde für eine realistische Benutzeranzahl herangezogen, in dem die aktiven Benutzer der MVG als Referenz hergenommen wurden. Diese beträgt 700.000 aktive Benutzer²⁸.

Für den Benchmark Test nimmt man eine Anzahl von 250.000 Benutzern an, die ihre Daten an die CEP auf einem Server senden. Mit Hilfe dieses Benchmark Tests kann die Skalierbarkeit der Architektur getestet werden, welche auch ein Teil der Fragen am Anfang dieses Kapitels darstellt.

Esper enthält ein Performanz-Kit, welches einen NIO-basierten Event Server und einen Client enthält welche herangezogen werden können, um den Performanztest durchzuführen. Das Kit ist in der Esper Bibliothek verfügbar. Der Esper Event Server erhält MarketData (Ticker, Preis, Volumen) Ströme welche von Remote Clients gesendet werden. Der Server hat per default 1000 registrierte Statements. Um jedoch die volle Power einer 32-bit und einer 64-bit Multicore Architektur zu erreichen, ist der Esper Event Server multithreaded. Die wichtigsten Parameter sind Latenz und Durchsatz. Latenz bezeichnet dabei die Dauer, zwischen dem verschicken zweier komplexer Events an die CEP. Der Durchsatz ist als das Maximum von Event/Sekunde definiert, welches die CEP ausführen kann, ohne Daten zu verlieren.

Der Benchmark Test für die vorliegende Arbeit wurde mit der Konfiguration aus **Tabelle 13** durchgeführt.

Spezifikation	
Betriebssystem für Esper Server und Client	Windows 7 Enterprise SP1 64-bit
Hardware für Esper Server und Client	1x Intel Core i5-3570 3.40 GHz (1core), 8GB RAM
Netzwerk	100 Mbit/s
JVM Version (Server und Client)	Java version „1.6.0_43“
JVM Parameter (Server)	-Xms2g -Xmx4g

Tabelle 13: Esper Event Server und Client Konfiguration

Der Server war so konfiguriert, dass es 100 Events/Sekunde von 2500 Clients erhalten konnte, welches zu einer Gesamtanzahl von 250.000 Events/Sekunde resultiert. Die Resultate der Simulation sind:

```

---Stats - engine (unit: ns)
Avg: 3995 #1160962
    0 < 5000 : 98,67% 99,67% #11454492
  5000 < 10000 : 0,84% 99,51% #9810
 10000 < 15000 : 0,17% 99,68% #1994
 15000 < 20000 : 0,09% 99,77% #1005
 20000 < 25000 : 0,02% 99,79% #261
 25000 < 50000 : 0,03% 99,83% #378
 50000 < 100000 : 0,05% 99,87% #553

```

²⁸ <http://www.mvg-mobil.de/fahrinfo>

100000	<	500000	:	0,10%	99,98%	#1200
500000	<	1000000	:	0,02%	99,99%	#206
1000000	<	2500000	:	0,00%	100,00%	#51
2500000	<	5000000	:	0,00%	100,00%	#10
5000000	<	more	:	0,00%	100,00%	#2
Throughput 250000 (active 0 pending 0)						

Dieser Durchlauf zeigt zum einen die durchschnittliche Latenz von Esper (Avg: 3995), zum anderen zeigt es die Gesamtanzahl der gesendeten Events (#1160962) und den Gesamtdurchsatz (Throughput 250000). Ein Latenzintervall wird in den ersten drei Spalten in ns definiert (z.B. 5000 < 10000). Die vierte Spalte beinhaltet den prozentualen Anteil von allen Events deren Latenz innerhalb des vorgesehenen Latenzintervalls lag. Die fünfte Spalte zeigt wie viel Prozent der Events weniger als der maximale Wert des Latenzintervalls benötigten und die letzte Spalte repräsentiert diesen Prozentsatz als eine Anzahl an Events.

Dieses Ergebnis zeigt, dass der Durchsatz bei 250.000 Events/Sekunde liegt und dass die durchschnittliche Esper Latenz bei 3995 ns (entspricht 4 µs) für 1.160.962 Events liegt. Weniger als 10 µs Latenz wurden mit 9810 Events erreicht, dies entspricht 99,51%. Latenz zwischen 5µs und 10µs wurde für die 0.84% aller Events in dem Intervall erreicht.

Eine mögliche Architektur für einen Backend-Sever könnte aus mehreren EPAs pro Domain bestehen. Eine Domain könnte Wetterinformationen, Staus usw. beinhalten. Jedes dieser EPAs müsste nur mit einigen Statements umgehen können, da jeder Benutzer maximal 1 Event/Sekunde sendet. Somit wären insgesamt ca. 250.000 Events/Sekunden, welche problemlos von einem EPA verarbeitet werden können (wie der oben angeführte Benchmark Test gezeigt hat).

5.5 ZUSAMMENFASSUNG

Mit diesem Kapitel 5 sollten folgende Fragestellungen beantwortet werden:

Kann CEP für intNAVs verwendet werden? Ja, ein Android Prototyp eines intNAVs wurde mit Hilfe von CEP im Rahmen dieser Arbeit entwickelt.

Was sind die Vor- und die Nachteile beim Benutzen von CEP für intNAVs? Der größte Vorteil liegt darin, komplexe Muster in großen Datenströmen in Echtzeit zu entdecken. Ein Nachteil ist, dass die Verwendung von CEP bei einem kleinen Eventstrom keine wirklichen Vorteile bringt.

Wie skalierbar ist solche eine Umsetzung? Ein simulierter Benchmark Test mit mehreren tausenden Benutzern zeigt, dass das System in Echtzeit auf mehrere tausende Events reagieren kann.

Der erste Aspekt wurde theoretisch bereits in Sektion 5.1.6 Zusammenfassung beantwortet, wo beinahe alle Kriterien um CEP zu verwenden erfüllt wurden. Der praktische Beweis folgte mit der Entwicklung des Java und des Android Prototypen, in Sektion 5.3.3.1 Java Prototyp und 5.3.3.2 Android Prototyp.

Die Vor- und Nachteile, welche im Rahmen der vorliegenden Arbeit entdeckt wurden, sind:

Vorteile, analysiert in Sektion 5.1.6 Zusammenfassung:

Einfache Implementierung von zeitlichen korrelierenden Events

Komplexe Events können im μs Bereich verarbeitet werden

EDA wurde entworfen um mit einer sehr großen Datenmenge umgehen zu können (Big Data)

Nachteile, analysiert in Kapitel 5.4:

Die Verwendung von CEP auf einem Client mit nur wenigen Events pro Sekunde ist nicht sehr hilfreich. In solch einem Fall, soll die Eventverarbeitung auf einem Server stattfinden. Dies ist natürlich nur dann sinnvoll, wenn immer eine Netzwerkverbindung zum Server existiert, da ansonsten keine Echtzeit Verarbeitung garantiert ist.

Die Skalierbarkeit wurde in Kapitel 5.4 untersucht. Es konnte gezeigt werden, dass Esper bei 250.000 Benutzern mit einer geringen Latenz und hohem Durchsatz arbeiten kann. Wenn die Anzahl an Benutzern steigt, sollten zusätzliche CEP Server mit mehreren EPAs hinzugefügt werden.

Es kann also zusammengefasst werden, dass CEP viele Vorteile für die Implementierung eines intNAVs mit sich bringt. Da sich in der Praxis die Anwendung einer intermodalen Navigation nicht auf einen Benutzer sondern auf mehrere hunderttausende Benutzer erstrecken kann, welche pro Sekunde mehrere tausende Events generieren und das intNAV von Echtzeitdaten lebt, ist die CEP Architektur für intNAVs besonders gut geeignet. Komplexe Events, wie Staus und Verspätungen, könnten dadurch effizient erfasst werden, um den Benutzer frühestmöglich auf Anomalien auf seiner gewählten Strecke hinzuweisen.

6 UMFRAGE UND NUTZERSTUDIE

Nachdem im vorangegangenen Kapitel 5 der Prototyp für ein intNAV präsentiert wurde, dient dieses Kapitel dazu, die Kriterien eines intNAVs, sowie die Funktionalität anhand von Umfragen und Nutzerstudien zu belegen. Dabei wird zunächst eine Umfrage präsentiert, anhand welcher Kriterien an ein intNAV abgeleitet werden können (1). Basierend auf diesen Kriterien, wurde der Prototyp entwickelt. Im zweiten Teil dieses Kapitels, werden die Kriterien anhand einer Nutzerstudie im intNAV hinsichtlich Funktionalität und Benutzerfreundlichkeit getestet (2). Abschließend werden die Ergebnisse der Umfrage und der Nutzerstudie in (3) zusammengefasst.

Das Ziel der vorliegenden Arbeit ist es ein intermodales Navigationssystem zu entwickeln, welches in Echtzeit auf aktive und passive Abweichungen innerhalb intermodaler Routen reagiert und den Benutzer, ähnlich wie bei einem traditionellen Navigationssystem, auf die Abweichungen hinweist und Alternativen vorschlägt. In den Kapiteln 3, 4 bis 5 wurden die technischen Grundlagen beschrieben, damit das Ziel der vorliegenden Arbeit, Erarbeiten eines funktionierenden intNAVs, erfüllt werden kann. In Kapitel 3 wurde der Prototyp auf seine Funktionsweise hin getestet, ob immer das korrekte Transportmittel vorhergesagt wird, in welchem sich der Benutzer befindet. In Kapitel 4 wurde isoliert die Erkennung der Abweichung getestet und in Kapitel 5 schließlich, wurde die Applikation zwar gesamthaft getestet, jedoch mit dem Fokus auf die Performanz der Anwendung.

Das vorliegende Kapitel dient dazu, die Anwendung (intNAV) hinsichtlich **Notwendigkeit**, **Funktionalität** und **Kriterien** für ein intNAV aber auch hinsichtlich **Zufriedenheit** beim Benutzer zu testen. Dazu wurden zwei Studien durchgeführt. In der ersten Studie wurde mittels einer Umfrage ermittelt, ob die Notwendigkeit für ein intNAV von den Probanden erkannt wird und auch, welche Kriterien an ein intNAV gestellt werden. Weiterhin wurde untersucht, wie die Personen über verschiedene Arten von Deviationen benachrichtigt werden möchten. Insbesondere sollte die Hypothese beantwortet werden, ob es einen Unterschied zwischen der Benachrichtigungsform von aktiven und passiven Deviationen gibt, also ob Personen anders über Abweichungen informiert werden möchten für die sie selber verantwortlich sind. Die Hypothese lautet, dass davon auszugehen ist, dass aktive Abweichungen absichtlich herbeigeführt werden und dass eine Person sehr wohl über die existierende Abweichung und auch die Folge davon Bescheid weiß und dass somit in solchen Fällen eine andere,

nicht so intensive, Benachrichtigung benötigt wird wie bei passiven Abweichungen. Als Neben aspekt dieser Umfrage wurde das Benutzerinterface sowie die Unterbrechungsart und -dauer untersucht.

Nachdem also die Umfrage wichtigen Inhalt für die gewünschte Funktionalität eines intNAVs geliefert hat wurde abschließend eine weitere Studie durchgeführt bei der es das Ziel war das entwickelte intNAV hinsichtlich seiner Funktionalität und Zufriedenheit beim Benutzer zu überprüfen. Dazu wurden mehrere Personen ausgewählt, die das intNAV über eine Woche hinweg in ihren Alltag integrieren sollen und es sowohl während der Benutzung, über einen Aufnahme-Button, als auch am Ende, mittels eines Interviews, bewertet haben. Eine der wichtigsten Faktoren hierbei war es herauszufinden ob die, in der Umfrage erkannten wichtigen Funktionalitäten, auch zur Zufriedenheit der Tester funktionieren und die Benutzer das vorliegende System akzeptieren. Sowohl die Umfrage als auch die Studie wurden nach der Entwicklung des intNAV Prototypen durchgeführt.

Nachfolgend werden der Aufbau und die Durchführung sowie die Ergebnisse der Umfrage und der Studie vorgestellt.

6.1 UMFRAGE

Wie eingangs erwähnt, hatte die Umfrage den Fokus, die Präferenzen der Benutzer hinsichtlich

Inhalt

Benachrichtigungsart und -dauer

Unterscheidung zwischen aktiven und passiven Störungen

Unterscheidung zwischen kontextbezogenen und nicht-kontextbezogenen Unterbrechungen

herauszufinden. Es sollten Indikatoren dafür identifiziert werden, wie ein intNAV aufgebaut sein soll und welche Informationen dieses enthalten soll. Das vorliegende Kapitel ist den Stichpunkten entsprechend aufgebaut, dies bedeutet, dass zunächst der gewünschte Inhalt bei einem intNAV bei den befragten Personen identifiziert wird um danach die aufgestellte Hypothese entweder zu bestätigen oder zu revidieren.

Die Hypothese lautet:

Bei der Art der Unterbrechung gibt es eine Benachrichtigungspräferenz abhängig davon es sich um eine aktive oder um eine passive Anomalie handelt. Dies bedeutet das davon auszugehen ist, dass bei einer aktiven Anomalie, also dann wenn der Benutzer selbst für die Störung auf der Route verantwortlich ist, eine andere Art der Unterbrechung gewünscht wird, als bei einer passiven Anomalie. Die Zusammenfassung der Umfrage kann im Appendix A nachvollzogen werden.

Nachfolgend werden Eckdaten zu den Teilnehmern sowie zu den bevorzugten Verkehrsmitteln dargestellt. Es wurden 34 Personen (davon 21 männlich und 13 weiblich) im Alter von 20 bis 52 Jahren (Durchschnitt 29,9 Jahre, Median: 27,5) befragt. In Hinblick auf die Haupttätigkeit der Probanden (BMW-Mitarbeiter und Studenten der Informatik) welches beide Männerdomänen sind, ist die Verteilung nicht unüblich. Es handelt sich zum größten Teil um Studenten und Angestellte. Die Hauptarbeitszeiten sind von 7 bzw. 8 Uhr bis 17 bzw. 18 Uhr.

Arbeitszeiten, die in etwa gleich liegen bedeuten ein erhöhtes Verkehrsaufkommen zu diesen Uhrzeiten. Ein kontextsensitives Navigationssystem, welches automatisch nach Störungen sucht, wenn sich der Benutzer auf den Weg in die Arbeit macht, würde den Nutzer somit zur richtigen Zeit unterstützen, ohne diesen dabei zu nerven.

6.1.1 Präferenzen bei der Verkehrsmittelwahl innerorts

Als eine der ersten wichtigen Erkenntnisse galt es herauszufinden, welches Verkehrsmittel Innerorts am meisten oder relativ häufig genutzt wird und ob daraus eine Tendenz zur Intermodalität abgeleitet werden kann. Natürlich kann über die Verteilung der am häufigsten verwendeten Verkehrsmittel innerhalb einer bestimmten Stadt keine Rückschlüsse über andere Städte oder andere Länder getroffen werden. Für die vorliegende Arbeit wurde exemplarisch die Stadt München ausgewählt. Wenn Klarheit über die Verteilung der Verkehrsmittel herrscht, kann die Sinnhaftigkeit nach einem intNAV beantwortet werden. Ausgehend von dieser Auswertung können weiterhin Rückschlüsse über die Information, welche angezeigt werden soll, gezogen werden. **Abbildung 58** stellt die Verteilung dar. Dass das Verkehrsmittel „zu Fuß“ relativ weit oben ist, überrascht wenig. Dies liegt daran, dass der Fußweg implizit bei jedem Verkehrsmittel dabei ist. Wenn das Auto verwendet wird, muss der Benutzer trotzdem zum Auto und, am Ende seiner Route, das Auto verlassen um zum Ziel zu kommen. Bei den ÖPNV muss der Benutzer ebenfalls zu Fuß zur Haltestelle gehen und auch beim Umsteigen in ein anderes ÖPNV muss der Benutzer ebenfalls den Weg zu Fuß zurücklegen. Somit wird dieses „Verkehrsmittel“ als redundant betrachtet und kann ignoriert werden.

Öffentliche Verkehrsmittel werden Innerorts (in München) am meisten genutzt. Da es das meistgenutzte Verkehrsmittel ist, werden Informationen rund um dieses Medium umso wichtiger. Dies bedeutet, dass die Anzeige von Störungen, Verspätungen und Alternativrouten hohe Priorität bei einem intNAV haben sollten.

Das Auto liegt bei der Umfrage knapp dahinter als das am zweithäufigsten genutzte Verkehrsmittel. Dies hängt damit zusammen, dass es zwar viele Vorteile bietet (Unabhängigkeit, keine Zwischenstopps, Komfort), aber auch Nachteile (im Berufsverkehr nicht schneller als ÖPNV, Teurer im Unterhalt). Da der Arbeitsweg und oft gefahrene Strecken dem Benutzer meist bekannt sind, sollte die Anzeige von Staus und Störungen im intNAV höhere Priorität genießen als die Route selbst.

Etwas abgeschlagen hinter beiden ist das Fahrrad. Der meistgenannte Vorteil ist, dass es am kostengünstigsten ist und der Benutzer in manchen Fällen schneller als mit dem Auto oder ÖPNV am Ziel ist. Der offensichtliche Nachteil ist der Zeit- und Kraftaufwand und dass es Wetterabhängig ist. Überraschenderweise werden Motorräder/ Roller nur von einer Minderheit Innerorts verwendet und das auch nicht regelmäßig. 85% der befragten hat entweder kein Motorrad/Roller oder benutzt es nie Innerorts. Das lässt auf den geringen Verteilungsgrad der Motorräder schließen.

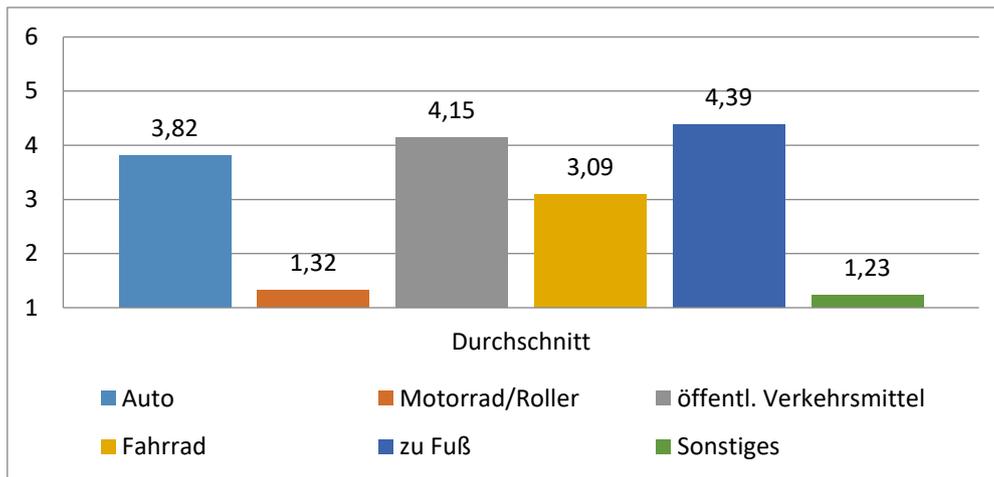


Abbildung 58: Meistverwendete Verkehrsmittel Innerorts in München (1 - nie bis 6 - jeden Tag)

Aus dieser ersten Erhebung lässt sich schließen, dass ÖPNV, Fahrrad und Auto die am meisten genutzten Verkehrsmittel Innerorts sind. Da, wie weiter oben schon beschrieben, die befragten Personen einen festen Tagesrhythmus haben, bei welchem die meisten zwischen sieben und acht Uhr morgens das Haus verlassen und zwischen 17 und 18 Uhr abends mit ihrer Arbeit fertig sind. Da sich der Weg in die Arbeit nicht ändert, ist ein weiteres Resultat der Studie, dass die Darstellung der Strecke für den Pendler uninteressant ist, da sich diese nicht ändert. Viel interessanter für den Anwender wäre es über Veränderungen auf seiner täglich befahrenden Strecke informiert zu werden. Somit sollte im intNAV die Strecke nicht den größten Platz einnehmen (vorausgesetzt diese ist bekannt).

Für das intNAV lassen sich folgende Schlüsse ziehen:

Die Verkehrsmittel „Auto“ und „ÖPNV“ müssen fehlerfrei von dem intNAV erfasst werden

Darstellung der Strecke nicht relevant, falls sich keine Änderung ergibt

Darstellung von aktiven und passiven Veränderungen sehr relevant

Darstellung von Alternativen, falls Änderung aufgetreten ist.

Nachdem die Notwendigkeit eines intNAVs bestätigt wurde, soll nun das Medium, über welches der Benutzer mit dem intNAV interagiert, ermittelt werden.

6.1.2 Mediennutzung und Erfahrung mit intNAV

Die Mediennutzung ist ein weiterer wichtiger Faktor für das hier entwickelte intNAV. Das intNAV wurde im Rahmen der vorliegenden Arbeit für mobile Endgeräte entwickelt. Die Umfrage soll nun dieses Vorgehen bestätigen. Wie schnell ersichtlich wurde, herrscht unter den Testpersonen fast eine hundertprozentige Abdeckung hinsichtlich der Erfahrung mit dem Umgang mit den Smartphones, wie in **Abbildung 59** zu sehen ist.

Da das intNAV einen nahezu flächendeckenden Standortzugriff benötigt wurde im nächsten Schritt untersucht in wie weit die Benutzer sich dadurch gestört fühlen, wenn sie bei der Verwendung des intNAVs das GPS permanent aktiviert haben müssen.

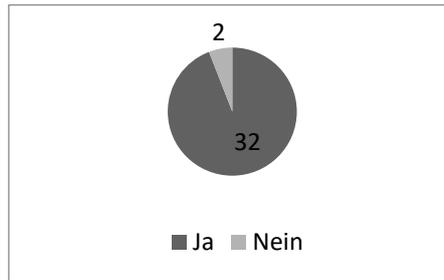


Abbildung 59: Erfahrung mit dem Umgang von mobilen Endgeräten

Dazu wurden den Benutzern im Rahmen der Umfrage zwei Fragen gestellt (vgl. **Tabelle 14**). Die Nutzer stehen der Tatsache, dass sie GPS aktiviert haben müssen, um das intNAV im vollen Umfang nutzen zu können neutral gegenüber. Jedoch wird der erhöhte Akkuverbrauch bemängelt.

Aussagen	0 [STÖRT MICH GAR NICHT]	1	2	3	4	5 [STÖRT MICH SEHR]	Mittelwert
Inwieweit stört es Sie, dass Sie das GPS aktiviert haben müssen, damit das intermodale Navigationssystem korrekt funktioniert?	35.29% (12)	23.53% (8)	5.88% (2)	26.47% (9)	5.88% (2)	2.94% (1)	2.53
Stört Sie ein erhöhter Akkuverbrauch in Folge der GPS Aktivierung?	2.94% (1)	11.76% (4)	26.47% (9)	20.59% (7)	14.71% (5)	23.53% (8)	4.03

Tabelle 14: Einstellung bzgl. GPS

Unmittelbar an diese Frage anknüpfend, geht es in der Umfrage weiterhin um die Erfahrungen der Probanden mit standortbezogenen Anwendungen und auch, ob sie darin ein Problem sehen, dass die Applikation Zugriff auf ihre standortbezogenen Daten hat. Die Umfrage ergab, dass die meisten Nutzer Erfahrung mit Standortdiensten haben (vgl. **Abbildung 60**). Alle Nutzer haben Erfahrung im Umgang mit Standortdiensten/Navigationssystemen/Routenplanern. Eine interessante Erkenntnis war, dass intermodale Navigationssysteme (intNAVs) wie sie in der vorliegenden Arbeit entwickelt werden, ca. der Hälfte der Befragten ein Begriff ist. Dies deutet darauf hin, dass das Thema zwar einigen bekannt aber noch nicht etabliert ist. Das könnte darauf zurückgeschlossen werden, dass es keine passable Lösung für intNAVs gibt, sodass sie noch nicht praktikabel sind. Im Gegensatz zu traditionellen Navigationssystemen, wie sie in einem Fahrzeug zu finden sind.

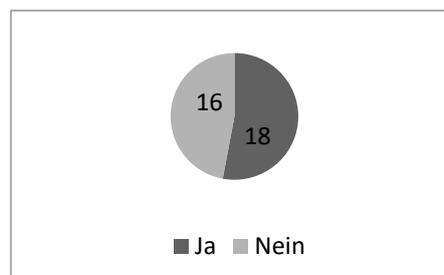


Abbildung 60: Erfahrung mit intermodalen Navigationssystemen

Für die vorliegende Arbeit folgt daraus, dass Smartphones (97%) und Navigationssysteme (100%) mittlerweile

in unserer Gesellschaft etabliert sind und praktisch jeder diese kennt und verwendet. Intermodale Navigationssysteme (auch auf Smartphones) sind noch wenig verbreitet und wenig bekannt. Aus dem mangelnden Bekanntheitsgrad intermodaler Navigationssysteme lässt sich eine potentielle Marktlücke schließen, was intNAVs auf Smartphones betrifft.

Nachdem im ersten Schritt die grundlegende Bereitschaft und Notwendigkeit von intNAVs sowie das entsprechende Medium auf welchem diese ausgeführt werden sollen ermittelt wurden, folgt nun im nächsten Schritt die inhaltliche Betrachtung des intNAVs, also darüber, welche Informationen die befragten Personen als nützlich in einem intNAV und welche als weniger nützlich empfinden.

6.1.3 Bevorzugte Funktionalität eines intNAVs

Ein Teil der Umfrage untersuchte, welche Funktionalität die Probanden bei einem intNAV als nützlich und welche als weniger nützlich empfinden. So soll die bestehende Funktionalität im intNAV entweder bestätigt oder revidiert werden können. Dazu bekamen die Probanden eine Liste von Aussagen bzw. Situationen genannt, welche sie auf einer Skala von (-5) bis (+5) bewerten sollten. Da die Umfrage durchgeführt wurde, als der Prototyp schon existierte, wurde die Umfrage so gestaltet, dass die bestehende Funktionalität kritisch hinterfragt wurde. Die **Tabelle 15** ist dabei ein Auszug aus der gesamten Befragung, welche im Appendix A nachgeschlagen werden kann. Dabei gehen die ersten vier Fragen auf die Funktionalität „Deviation“ (vgl. Kapitel 4) ein und drei Fragen auf die „Transportmittelerkennung“ (vgl. Kapitel 3).

Dabei ist die Skala für die Bewertung folgendermaßen zu lesen:

Sehr negativ (-5): „*Ich empfinde das als schlecht.*“, „*Es stört mich sehr.*“, „*Ich bin der Aussage gegenüber sehr negativ eingestellt*“

Neutral (0): „*Ich empfinde diesen Punkt weder als positiv, noch als negativ.*“, „*Ich sehe sowohl negatives, als auch positives*“

Sehr positiv (+5): „*Ich empfinde das als gut/nützlich.*“ „*Das ist eine tolle Funktion*“ „*Ich bin der Aussage gegenüber sehr positiv eingestellt.*“

Um in den jeweiligen Tabellen die letzte Spalte („Median“) ausrechnen zu können, werden den Bewertungen jeweils natürliche Zahlen zugeordnet. So erhält die Bewertung (-5) die natürliche Zahl 1, die Bewertung (-4) die Zahl 2, bis zur Bewertung (+5), welche natürliche Zahl 11 erhält.

Frage:

Im Folgenden werden Ihnen Aussagen vorgelegt, die sich auf das intermodale Navigationssystem beziehen. Bewerten Sie nun diese Aussagen, inwiefern Sie eine positive, neutrale oder negative Haltung bei Ihnen auslösen

Aussagen	-5 (1)	-4 (2)	-3 (3)	-2 (4)	-1 (5)	0 (6)	1 (7)	2 (8)	3 (9)	4 (10)	5 (11)	Median (1-11)
Das intermodale Navigationssystem sendet Benachrichtigungen, wann die Route gestartet werden soll.	0.00% (0)	0.00% (0)	0.00% (0)	2.94% (1)	0.00% (0)	14.71% (5)	11.76% (4)	20.59% (7)	20.59% (7)	8.82% (3)	20.59% (7)	8.47
Das intermodale Navigationssystem ändert bei einer abweichenden Startposition die Startzeit und weist den Nutzer darauf hin.	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	2.94% (1)	2.94% (1)	2.94% (1)	17.65% (6)	17.65% (6)	20.59% (7)	35.29% (12)	9.47
Das intermodale Navigationssystem meldet, wenn auf der Route ungeplante Störungen auftreten, die Einfluss auf die Ankunftszeit haben. ²⁹	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	3.03% (1)	0.00% (0)	18.18% (6)	24.24% (8)	21.21% (7)	33.33% (11)	9.61
Das intermodale Navigationssystem erkennt, dass der Nutzer noch nicht losgegangen ist und informiert ihn sofort	0.00% (0)	0.00% (0)	0.00% (0)	5.88% (2)	0.00% (0)	17.65% (6)	5.88% (2)	11.76% (4)	20.59% (7)	11.76% (4)	26.47% (9)	8.59
Das intermodale Navigationssystem erkennt mit welchem Verkehrsmittel man unterwegs ist.	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	2.94% (1)	2.94% (1)	11.76% (4)	17.65% (6)	26.47% (9)	38.24% (13)	9.76
Das intermodale Navigationssystem erkennt, dass man in das falsche öffentliche Verkehrsmittel gestiegen ist.	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	2.94% (1)	2.94% (1)	17.65% (6)	29.41% (10)	14.71% (5)	32.35% (11)	9.47
Das intermodale Navigationssystem benutzt alle zur Verfügung stehenden Verkehrsmittel, um die Route zu berechnen.	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	2.94% (1)	8.82% (3)	2.94% (1)	5.88% (2)	17.65% (6)	14.71% (5)	47.06% (16)	9.59
Das intermodale Navigationssystem schlägt Alternativrouten vor.	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	2.94% (1)	11.76% (4)	29.41% (10)	20.59% (7)	35.29% (12)	9.74

Tabelle 15: Kriterien eines intNAVs (Auszug aus Appendix A)

Wie der **Tabelle 15** nun entnommen werden kann, wird die entwickelte Funktionalität bzgl. Deviationenerkennung als auch der Transportmittelerkennung auch von den befragten Personen unterstützt. Es können minimale Präferenzen identifiziert werden. So ist die Transportmittelerkennung die am wichtigsten

²⁹ 33 gültige Bewertungen anstelle von 34 konnten bei dieser Frage gezählt werden.

betrachtete Funktionalität eines intNAVs (Median von 9.76), dicht gefolgt von den Alternativroutenvorschlägen (Median von 9.74) und der Störungserkennung (Median von 9.61).

Wenn die Tabelle nun genauer betrachtet wird, fällt auf, dass Funktionalitäten wie

Das intermodale Navigationssystem meldet, wenn auf der Route ungeplante Störungen auftreten, die Einfluss auf die Ankunftszeit haben.

Das intermodale Navigationssystem erkennt mit welchem Verkehrsmittel man unterwegs ist.

Das intermodale Navigationssystem erkennt, dass man in das falsche öffentliche Verkehrsmittel gestiegen ist.

Das intermodale Navigationssystem schlägt Alternativrouten vor.

keine einzige Stimme bei den negativen Bewertungen ((-5) bis (-1)) erhalten haben. Bei diesen vier Aussagen, ist eine Häufung insbesondere bei den höheren Bewertung festzustellen, also bei (+3) bis (+5). Dies ist insofern interessant, dass genau diese Funktionalitäten auch Fokus der vorliegenden Arbeit sind.

Da die Akzeptanz bzw. Nützlichkeit von der Deviationenerkennung, also der aktiven und passiven Störungen, in der Umfrage klar hervorkommt, widmet sich nun das nächste Unterkapitel der eingangs in diesem Kapitel geäußerten Hypothese, nämlich ob die Probanden einen Unterschied in der Benachrichtigungsart von aktiven und passiven Störungen sehen.

6.1.4 Präferenzen bzgl. Inhalt und Stärke der Benachrichtigung bei Anomalien

In Kapitel 4 wird ausführlich über die Unterscheidung zwischen aktiven und passiven Störungen berichtet. Aktive Störungen sind solche, für welche der Benutzer **selbst verantwortlich** ist und/oder diese **bewusst** ausübt, also beispielsweise

zu spät Losgehen

in das falsche Transportmittel steigen

anders als vorgesehen abbiegen

Passive Störungen werden hingegen von der **Umwelt** und nicht vom Benutzer initiiert, beispielsweise

Verspätungen

Ausfälle

Staus

Nun lautet die aufgestellt Hypothese, dass der Benutzer, wenn er selbständig für eine Störung verantwortlich ist, weniger penetrant darüber benachrichtigt werden möchte, als wenn diese Störung durch die Umwelt initiiert ist und der Benutzer damit nicht rechnen konnte. Man geht bei dieser Hypothese davon aus, dass der Benutzer schnell von den Benachrichtigungen genervt sein könnte, weil er ja weiß, dass er diese Störung initiiert hat. Er ist sich vermutlich auch über die Konsequenzen im Klaren und hat schon einen Plan B parat. Um diese Hypothese nun zu überprüfen, wurden folgende Punkte untersucht:

Welche Inhalte sollen bei einer erkannten Störung angezeigt werden?

Welche Benachrichtigungsintensität wird bevorzugt?

Spielt die Situation, in welcher sich der Benutzer befindet, eine Rolle, auf welcher Art Benachrichtigungen kommuniziert werden sollen?

Der Hintergrund dieser Fragestellungen ist, dass ein Indiz daraus abgeleitet werden kann, ob der Benutzer einen anderen Umgang zwischen aktiven und passiven Störungen wünscht.

Inhalt

Zunächst werden die Präferenzen der Benutzer zum Inhalt in Bezug auf die aktiven Störungen untersucht, wie in **Tabelle 16** zu sehen ist. Dabei lautete die Frage in der Umfrage folgendermaßen:

Frage: Neben dem Grund für die Benachrichtigung (aktive Störung liegt vor) können weitere Informationen angezeigt werden. Nennen Sie solche, die Sie als Nutzer gerne auf einen Blick sehen würden.

	0 (1)	1 (2)	2 (3)	3 (4)	4 (5)	5 (6)	Median (1-6)
Klassifizierung - um was für eine Störung es sich handelt (z.B. befindet sich im falschen öffentlichen Verkehrsmittel)	0.00% (0)	0.00% (0)	2.94% (1)	5.88% (2)	26.47% (9)	64.71% (22)	5.53
Alternativen/ Aktionsmöglichkeiten – Benutzer wird über alternative Routen informiert (z.B. anderen Bus nehmen)	0.00% (0)	2.94% (1)	8.82% (3)	20.59% (7)	23.53% (8)	44.12% (15)	4.97
Konsequenzen – Auswirkungen, wenn die Route nicht eingehalten wird (z.B. U-Bahn verpasst, nächste in 30 Minuten)	0.00% (0)	11.76% (4)	11.76% (4)	23.53% (8)	32.35% (11)	20.59% (7)	4.38

Tabelle 16: Ergebnisse der Befragung zu Informationen bei erkannten aktiven Störungen (Auszug aus Appendix A)

Anhand der Befragung lässt sich zunächst feststellen, dass die Benutzer alle drei Informationen als hilfreich empfinden, jedoch mit folgender Abstufung: „Klassifizierung“ > „Alternativen“ > „Konsequenzen“. Dies zeigt deutlich, dass die Benutzer, obwohl es sich um eine aktive Störung handelt, an der „Klassifizierung“, also an dem Grund für die Störung Interesse haben. Dies ist gleichzeitig ein Teil der eingangs aufgestellten Hypothese, in der angenommen wurde, dass der Benutzer den Grund für die aktive Abweichung kennt. Laut dieser Umfrage, zeigen die Probanden gerade an dieser Information das meiste Interesse. Somit kann ein Teil der Hypothese bereits revidiert werden.

Weniger erstaunlich ist, dass die Information „Konsequenzen“ verhältnismäßig als am unwichtigsten empfunden wird. Die Begründung hierfür ist ziemlich naheliegend, denn die Konsequenz einer aktiven (aber auch passiven) Störung ist immer die gleiche, nämlich, dass die geplante Route nicht eingehalten werden kann und das Ziel vermutlich nicht, wie erwartet, erreicht werden kann.

Die gleiche Frage wurde ebenfalls bei passiven Störungen gestellt, mit folgender Verteilung (vgl. **Tabelle 17**):

Frage: Neben dem Grund für die Benachrichtigung (passive Störung liegt vor) können weitere Informationen angezeigt werden. Nennen Sie solche, die Sie als Nutzer gerne auf einen Blick sehen würden.

	0 (1)	1 (2)	2 (3)	3 (4)	4 (5)	5 (6)	Median (1-6)
Klassifizierung - um was für eine Störung es sich handelt (z.B. befindet sich im falschen öffentlichen Verkehrsmittel)	0.00% (0)	2.94% (1)	0.00% (0)	14.71% (5)	20.59% (7)	61.76% (21)	5.38
Alternativen/ Aktionsmöglichkeiten – Benutzer wird über alternative Routen informiert (z.B. anderen Bus nehmen)	0.00% (0)	0.00% (0)	8.82% (3)	17.65% (6)	23.53% (8)	50.00% (17)	5.15
Konsequenzen – Auswirkungen, wenn die Route nicht eingehalten wird (z.B. U-Bahn verpasst, nächste in 30 Minuten)	0.00% (0)	5.88% (2)	14.71% (5)	14.71% (5)	29.41% (10)	35.29% (12)	4.74

Tabelle 17: Ergebnisse der Befragung zu Informationen bei erkannten passiven Störungen (Auszug aus Appendix A)

Die Darstellung weiterer Informationen ist ähnlich zu den aktiven Störungen. Auch hier erachten die Probanden alle drei Informationen als wichtig, wobei auch hier die gleiche Aufteilung vorzufinden ist: „Klassifizierung“ > „Alternativen“ > „Konsequenzen“. „Konsequenzen“ und „Alternativen“ erscheinen den Benutzern bei passiven Störungen etwas wichtiger, da Konsequenzen nicht klar ersichtlich sind und Lösungen wichtiger eingeordnet werden. Die **Abbildung 61** zeigt eine graphische Gegenüberstellung der erwähnten Ergebnisse für aktive und passive Störungen.

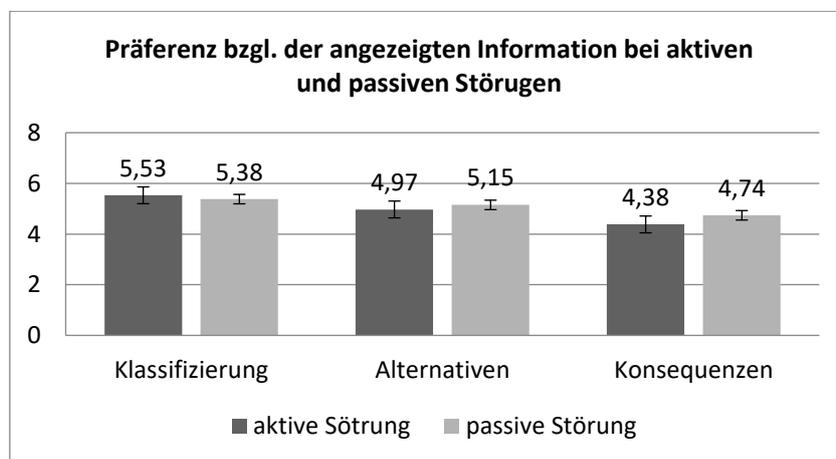


Abbildung 61: Gegenüberstellung der Umfrageergebnisse bzgl. der Anzeige des Inhalts von Störungen (0 – unwichtig bis 6 – sehr wichtig)

Benachrichtigungsintensität

Neben dem Inhalt, welchen die Benutzer bei aktiven und passiven Störungen als nützlich empfinden, sollte in der Umfrage auch die Benachrichtigungsdauer und Benachrichtigungsintensität ermittelt werden. Mit Hilfe dieser Fragestellung soll ein Indiz gefunden werden, dass sich der Benutzer bei aktiven Störungen eine weniger intensive

Benachrichtigungsart wünscht als bei einer passiven Abweichung. Wenn die Umfrageergebnisse zeigen, dass bei passiven Abweichungen, der Benutzer eine längere und intensivere Benachrichtigungsart wünscht, wäre die eingangs in diesem Kapitel vorgestellte Hypothese bestätigt.

Wie sich anhand der Umfrage (vgl. **Tabelle 18**) feststellen lässt, wünschen sich nur 30% der Befragten eine dauerhafte Benachrichtigung, bis die Störung nicht mehr besteht. Interessanterweise ist das Ergebnis diesbezüglich sowohl bei aktiven als auch bei passiven Benachrichtigungen fast das gleiche, es unterscheidet sich lediglich um eine Stimme, welche sich bei aktiven Abweichungen eine penetrantere Art der Benachrichtigung wünscht.

Frage: Möchten Sie so lange Benachrichtigungen erhalten bis die Störung nicht mehr besteht?

	Anzahl	Prozent
Ja	11	32.35%
Nein	23	67.65%

	Anzahl	Prozent
Ja	10	29.41%
Nein	24	70.59%

Tabelle 18: Umfrageergebnisse bzgl. Benachrichtigungsdauer bei aktiven (links) und passiven (rechts) Störungen (Auszug aus Appendix A)

Dabei gab es eine zusätzliche Auswertung, wie häufig eine Benachrichtigung gesendet werden sollte. Die Auswahlmöglichkeit reichte dabei von 10 Sekunden über jede Minute bis zu 5 Minuten (vgl. **Tabelle 19**). Die Mehrheit der Befragten (46.67% vs. 53.33%) hält alle 5 Minuten bei einer aktiven Störung aber auch bei einer passiven Störung als vollkommen ausreichend. Die Mehrzahl der Nutzer will eine Benachrichtigung, um auf die Störung hingewiesen zu werden, aber nicht periodisch, da die Nachrichten keinen zusätzlichen Mehrwert hätten.

Frage: Wenn ja, in welchen Intervallen sollten die Benachrichtigungen eintreffen?

	Anzahl	Prozent
Alle 10 Sekunden	1	6.67%
Alle 30 Sekunden	3	20.00%
Jede Minute	2	13.34%
Alle 2 Minuten	2	13.34%
Alle 5 Minuten	7	46.67%

	Anzahl	Prozent
Alle 10 Sekunden	1	6.67%
Alle 30 Sekunden	1	6.67%
Jede Minute	1	6.67%
Alle 2 Minuten	4	26.67%
Alle 5 Minuten	8	53.33%

Tabelle 19: Umfrageergebnisse bzgl. intervallartiger Benachrichtigungsart bei aktiven (links) und passiven (rechts) Störungen (Auszug aus Appendix A)

Bei der direkten Gegenüberstellung, (vgl. **Abbildung 62**) fällt auf, dass die Tendenz bei der Benachrichtigungsart bei den Benutzern grundsätzlich die gleiche ist. Die meisten wollen eher alle 5 Minuten über eine eingetretene Störung informiert werden. Dies ist sowohl bei der aktiven als auch bei der passiven Störung identisch. Jedoch ist auffällig, dass sich die Zahlen bei 30 Sekunden und 1 Minute unterscheiden. Wengleich bei passiven Abweichungen die Kurve sehr eindeutig ist, nämlich keine häufige Benachrichtigung, fällt bei der aktiven Abweichung ein leichter Knick bei 30 Sekunden und 1 Minute auf. Dies kann doch so interpretiert werden, dass es einigen Benutzern wichtig erscheint, bei aktiven Abweichungen häufiger

Benachrichtigungen zu erhalten als bei passiven. Jedoch stimmt die Mehrheit der Befragten bei den 5 Minuten überein, sodass sich hierbei die eingangs erwähnte Hypothese nicht bestätigen lässt.

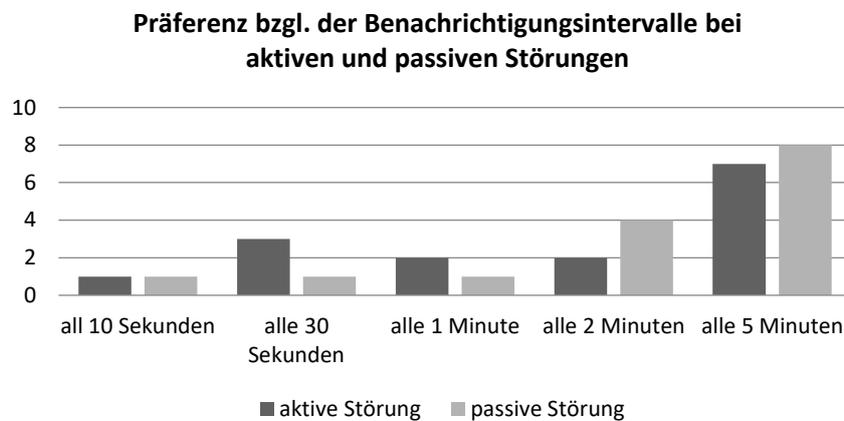


Abbildung 62: Gegenüberstellung der Präferenz bzgl. der Benachrichtigungsintervalle von aktiven und passiven Störungen

Situationen

Schließlich soll anhand von Situationen herausgefunden werden, ob die Probanden in unterschiedlichen Situationen eine unterschiedliche Intensität der Benachrichtigungen bei aktiven und passiven Störungen wünschen. Dazu wurden elf Situationen ausgewählt, bei welchen die Probanden angeben sollen, wie sehr sie sich dabei von einer Benachrichtigung gestört fühlen würden basierend auf aktiven und passiven Deviationen. Hier wurde zusätzlich untersucht, ob es innerhalb der jeweiligen Situationen einen Unterschied zwischen einmaliger Benachrichtigung bei einer aufgetretenen Störung einer mehrmaligen Benachrichtigung gibt. Insgesamt wurde also untersucht, ob es situationsabhängig einen Unterschied in der Benachrichtigungsintensität (einmalig vs. Mehrmalig) und bei der Art der Störung (aktiv und passiv). Die **Abbildung 63** stellt die Umfrageergebnisse aus Appendix A graphisch gegenüber.

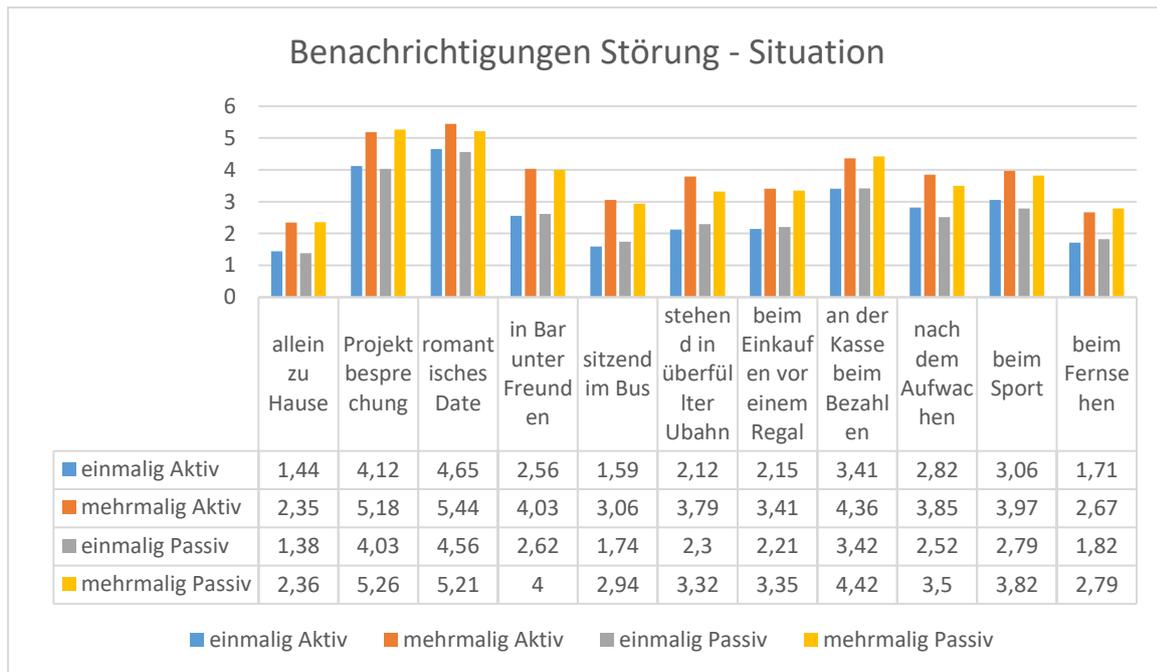


Abbildung 63: Situationsabhängig Benachrichtigung 1 – nicht störend ... 6 – sehr störend)

Aus der Umfrage und somit der **Abbildung 63** lassen sich folgende Schlüsse ziehen:

einmalige Benachrichtigungen werden von den Benutzern über sämtliche Situationen hinweg als besser empfunden als mehrmalige.

Dabei wurde der Wilcoxon-Vorzeichen-Rang-Test und der Wilcoxon-Mann-Whitney-Test verwendet, um zu überprüfen, ob sich einmalig-mehrmalig (jeweils aktiv und passiv) unterscheiden. Der Unterschied ist signifikant ($p < 0.01$).

Es gibt keinen Unterschied bei aktiven oder passiven Störungen.

Es wurde mit beiden Tests überprüft, ob es einen Unterschied zwischen aktiv oder passiv gibt (jeweils einmalig und mehrmalig). Es konnte kein signifikanter Unterschied festgestellt werden ($p > 0.8$).

Somit lässt sich zusammenfassen, dass die eingangs in diesem Kapitel erwähnte Hypothese, der Benutzer erwartet bei einer aktiven Störung eine andere Art der Benachrichtigung als bei einer passiven, nicht bestätigt werden konnte. Es konnte in der Umfrage kein signifikanter Unterschied zwischen aktiv und passiv festgestellt werden.

6.1.5 Zusammenfassung

Bei der Umfrage kam keine eindeutige Präferenz der Benutzer hinsichtlich *Inhalt*, *Benachrichtigungsart* und *Situation* bezogen auf aktive und passive Störungen heraus, also kann die aufgestellte Hypothese nicht bestätigt werden. Dafür kommen unterschiedliche Gründe in Frage. Zum einen ist der Ausgang, also egal ob eine aktive oder passive Verspätung vorliegt, der gleiche, nämlich, dass die geplante Route nicht eingehalten werden kann und somit eine Alternativroute gewünscht ist. Zum anderen kann mittels einer Anwendung nicht festgestellt

werden, ob die aktive Störung beabsichtigt war oder nicht. Ein Benutzer kann einfach die Zeit vergessen haben und deshalb nicht pünktlich losgehen. Der Benutzer kann aber aufgrund von Unwissenheit in ein „falsches“ Transportmittel eingestiegen sein. Ein Benutzer kann sich zwar bewusst dafür entscheiden, eine andere Strecke zu fahren, weil dieser beispielsweise noch tanken möchte er kann aber auch unbewusst falsch abgebogen sein und fährt somit eine andere (falsche) Strecke. Dann ist es keine beabsichtigte Störung und der Benutzer würde sich eine ähnliche Benachrichtigungsart wie bei der passiven Störung wünschen. Aber selbst wenn die Intention für eine Abweichung erkannt wird, ändert es laut der Umfrage trotzdem nichts daran, wie die Benutzer die Benachrichtigungsart sehen, nämlich sowohl bei aktiven als auch bei passiven Anomalien gleich.

6.2 NUTZERSTUDIE

Obwohl die einzelnen Teilfunktionalitäten bereits jeweils für sich genommen getestet wurden sollte mit Hilfe einer Nutzerstudie die Gesamtfunktionalität und die Akzeptanz des hier entwickelten intNAVs ermittelt werden.

6.2.1 Ziel der Nutzerstudie

Da im Rahmen der hier vorliegenden Arbeit ein Prototyp mit einer, bis zu diesem Zeitpunkt nicht vergleichbaren Funktionalität entwickelt wurde, bestand das Hauptziel der Nutzerstudie zunächst im Testen der Funktionalität des Prototyps. Genauso wichtig wie die Funktionalität ist jedoch auch die Akzeptanz der hierbei entwickelten Funktionen, da solch eine Art von Navigationssystemen bis zu diesem Zeitpunkt nicht vorliegt. Die Unterteilung der beiden Bereiche in die entsprechenden Unterfragen wird nachfolgend genauer betrachtet.

FUNKTIONALITÄT

Die Hauptfunktionalitäten des in dieser Thesis entwickelten intNAVs bestehen aus der Transportmittelerkennung in Echtzeit und der Erkennung von Anomalien. Diese, sowie die Performanz der Applikation sollen im Rahmen der vorliegenden Nutzerstudie getestet werden.

Transportmittelerkennung in Echtzeit

Die Transportmittelerkennung (vgl. Kapitel 3) bildet einen der zwei Hauptfunktionalitäten des hier vorgestellten Prototypen. Dabei soll ermittelt werden, ob während der Benutzung der Applikation, jeweils immer das korrekte Transportmittel erkannt wurde mit welchem der Benutzer unterwegs war.

Erkennung von aktiven Anomalien

Die Abweichenerkennung bildet die zweite wesentliche Funktion eines intNAVs. Abweichenerkennung besteht aus der Erkennung von aktiven und passiven Abweichungen (vgl. Kapitel 4). Die Erkennung von aktiven Abweichungen ist dabei der wesentliche Beitrag der hier vorgestellten Thesis. Wie bereits erwähnt wurde, kann jede Störung (egal ob aktiv oder passiv) eine Auswirkung auf die gesamte Route haben. Die Herausforderung, und auch der Mehrwert, des hier vorgestellten Prototypen ist, die Verknüpfung von Störungen innerhalb einer intermodalen Route zu erfassen.

Performanz der Applikation

Wie in Kapitel 5 beschrieben steht, spielt die Performanz einer mobilen Anwendung eine sehr wichtige Rolle. Kritisch sind dabei Anwendungen, die auf Echtzeit-Daten angewiesen sind. Bei diesen spielt die Performanz eine sehr wichtige Rolle. Die Benutzer benötigen aktuelle Daten und keine veralteten.

AKZEPTANZ DER APPLIKATION

Eine Applikation besteht nicht nur in der einwandfreien Funktionalität, sondern auch in wie weit die Interaktionsmöglichkeiten zwischen dem Benutzer und der Anwendung auf Akzeptanz stoßen. Um diesen Punkt zu messen wurde die Kommunikation von Störungen, Sinnhaftigkeit von Empfehlungen, die Informationsdichte sowie die Wiederkehrwahrscheinlichkeit untersucht. Nachfolgend werden alle vier Punkte nochmal genauer beschrieben.

Kommunikation von Störungen

Neben der Erkennung von Anomalien (Funktionalität) ist auch die angemessene Kommunikation dessen ein wichtiger Indikator darüber, wie hoch die Akzeptanz der Applikation ausfällt. Dabei wird die frühzeitige Kommunikation von aktiven und passiven Störungen untersucht sowie ob diese Störungen in einer angemessenen Art und Weise kommuniziert wurden.

Empfehlungen

Wenn aktive Störungen erkannt wurden, wurden zusätzlich Empfehlungen ausgegeben, wie beispielsweise „wenn du dich beeilst, schaffst du die geplante Route noch rechtzeitig.“ Als Teil der Studie soll untersucht werden, ob die Empfehlungen korrekt waren und ob sie geholfen haben.

Informationsdichte

Da die zur Verfügung stehende Fläche, auf welcher die gesamten Informationen angezeigt werden können, bei einem Smartphone gering ist, soll in der Studie untersucht werden, ob bei der intermodalen Navigation immer die relevante Information zur Verfügung stand oder ob wichtige Information gefehlt hat.

Wiederkehrwahrscheinlichkeit

Die Wiederkehrwahrscheinlichkeit spiegelt den größten Grad der Akzeptanz des entwickelten Prototypen wieder. Wenn dieser Punkt von den meisten Studienteilnehmern positiv bewertet wird, kann davon ausgegangen werden, dass die Anwendung ihren versprochenen Zweck erfüllt hat. Folgende drei Punkte fallen da drunter: „Die Applikation hat mich gut unterstützt“, „*Ich würde sie wiederverwenden*“ und „*Ich würde die Applikation weiterempfehlen*“.

Der Aufbau, der Ablauf sowie die Ergebnisse der Nutzerstudie, um die genannten Punkte „Funktionalität“ und „Akzeptanz“ zu testen, werden nachfolgend beschrieben.

6.2.2 Aufbau der Studie

An der Studie haben insgesamt 64 Personen (38 männlich und 26 weiblich) teilgenommen, im Alter von 19 bis 26 Jahren. Bei den Personen handelt es sich dabei um Bachelor- und Masterstudenten der Medieninformatik der Ludwig-Maximilians-Universität München. Alle Personen haben Erfahrung in der Programmierung und Evaluierung von Android-Anwendungen und erweiterte Kenntnisse in Mensch-Computer-Interaktion Design.

Dieser Personenkreis wurde ausgewählt, um innerhalb kürzester Zeit eine große Menge an Feedback bezüglich der Applikationen erhalten zu können. Der Testzeitraum betrug für alle Testpersonen sechs Tage (Start: Mittwoch 23.09.2015, Ende: Montag der 28.09.2015) und deckte somit auch das Wochenende ab. Der Aufruf zur Studie wurde dabei über Facebook in verschiedenen Gruppen der Studenten der Medieninformatik kommuniziert. Zu Beginn der Studie wurde in Gruppen die Funktionalität der Applikation erläutert, sowie eine Einweisung gegeben und die Feedbackoptionen besprochen (vgl. Appendix B). Interessenten konnten sich dazu an acht zur Verfügung stehenden Terminen anmelden, welche ebenfalls über Facebook in den entsprechenden Gruppen kommuniziert wurden. Die Teilnehmer sollten die Applikation installieren und führten den Prototyp auf ihren eigenen Smartphones aus. Zusätzlich wurde jedem Teilnehmer die tempv16.db auf die SD-Karte installiert, damit die Transportmittelerkennung funktioniert. Die Android-Versionen reichten dabei von 4.1.1 bis 5.0.2.

Um nun die Nutzerstudie zu evaluieren, wurde das Feedback der Benutzer auf folgenden drei Wegen erhoben:

- Möglichkeit einer „in-App“-Feedback Abgabe mittels eines Aufnahme-Knopfs.
- Führen eines Tagebuchs.
- Teilnahme an einem Abschlussinterview.

Während der gesamten Dauer von 6 Tagen konnten die Testpersonen zu jeder Zeit, wenn sie die Applikation verwenden, auf einen Aufnahmeknopf innerhalb der Anwendung klicken und sofort lossprechen. Das Feedback, welches sie auf diese Weise mitteilten, wurde als mp3-Dateien per E-Mail an den Testleiter versendet, in dem Moment wo die Testperson auf „Absenden“ geklickt hat. Diese Feedbacks wurden zusätzlich verwendet um mehr über die Akzeptanz der Benutzer zu der Applikation zu verstehen oder ggf. Fehler in der Applikation zu identifizieren.

Da der Nutzer die Applikation als Navigationssystem benutzt ist er immer unterwegs. Unter diesen Bedingungen ist kein Labortest zu empfehlen. Das Tagebuch bietet die Möglichkeit, Stimmungen und Meinungen, Aktivitäten und Erlebnisse der Nutzer einfangen zu können, ohne selbst als Beobachter direkt vor Ort zu sein. Dabei wurde ein Online-Fragebogen erstellt, wobei jeder Nutzer einen Code erhalten hat, welchen dieser zur Zuordnung beim Start des Fragebogens eingeben musste. Bei dem Fragebogen mussten die Benutzer Fragen zur Funktionalität und Akzeptanz der Applikation beantworten (siehe Appendix B).

Das Abschlussinterview wurde nach den sechs Tagen von den Benutzern online ausgefüllt. Dabei sollte nochmal zusätzlich zu der täglichen Bestandsaufnahme ein Gesamtbild von den Benutzern über die Funktionalität der Applikation gewonnen werden (siehe Appendix B). Während bei der Tagebuch-Form Fragen zur Funktionalität und Akzeptanz gestellt wurden (bspw.: *„Die verwendeten Transportmittel wurden von dem intNAV korrekt erkannt (1-7).“*), diente das Abschlussinterview dazu, das Meinungsbild zu der Applikation zu identifizieren (bspw.: *„Meine Erwartungen an die Applikationen wurden enttäuscht, erfüllt oder übertroffen.“*). In beiden Fragebögen steht jedoch sowohl die Funktionalität als auch die Akzeptanz im Vordergrund.

6.2.3 Auswertung

Die Ergebnisse der im vorangehenden Kapitel 6.2.2 erwähnten Punkte zum Testen der Funktionalität sowie der Akzeptanz des Prototypen und somit des im Rahmen der vorliegenden Arbeit entwickelten intNAVs werden nachfolgend präsentiert um im nächsten Kapitel 6.3 die Ergebnisse zu diskutieren. Dabei werden die Ergebnisse aus den beiden Abfrageformen „Tagebuch“ sowie „Abschlussinterview“ getrennt betrachtet. Die Untersuchungsart mittels der Aufnahmefunktion lässt keine einheitliche Verwertung zu, da die Ergebnisse sehr individuell sind. Diese Untersuchungsart wurde eher zu Fehlerbehebungs-Zwecken verwendet.

TAGEBUCH

Nachfolgend werden die Fragen zu den sieben Untersuchungspunkten (vgl. Kapitel 6.2.2) aufgelistet, sowie die Verteilung der Bewertung auf einer Skala von 1-7 vorgestellt. Dabei bedeutet (1) auf der Skala „stimme überhaupt nicht zu“ und (7) „stimme voll und ganz zu“. (4) ist dabei als „neutral“ zu sehen. Die Verteilung der jeweiligen Ergebnisse entspricht jeweils der Summe der entsprechenden Bewertungen über alle 64 Studienteilnehmer. Als wichtige Mess- und Vergleichsgröße wird der Mittelwert verwendet, welcher den Durchschnitt über sämtliche Verteilungen bildet. Dabei gilt, je größer der Mittelwert desto besser wurde der entsprechende Punkt bewertet. Dabei kann der Mittelwert ein Maximum von „7“ erreichen. In **Tabelle 20** ist das Ergebnis der Auswertung über die sieben Kategorien dargestellt. Dabei kann zum einen der Mittelwert entnommen werden, sowie die prozentuale Verteilung der eher positiven und der eher negativen Bewertungen pro Kategorie. Nachfolgend ist die Aufschlüsselung der einzelnen Zahlen in den entsprechenden Messgrößen dargestellt.

Messgrößen	Mittelwert	Positiv [%]	Negativ [%]
Transportmittelerkennung in Echtzeit	5,20	78,13	12,50
Erkennung von aktiven Anomalien	4,28	40,63	21,88
Performanz der Applikation	5,83	93,75	1,56
Kommunikation von Störungen	4,28	43,75	25,00
Empfehlungen	4,47	54,69	26,56
Informationsdichte	3,45	12,50	54,69
Wiederkehrwahrscheinlichkeit	5,42	81,25	4,69

Tabelle 20: Bewertung der sieben Messgrößen mit einer positiven und negativen Tendenz

Transportmittelerkennung in Echtzeit

Die erste Frage welche die Teilnehmer der Studie in dem Tagebuch beantworten sollten, dreht sich um die Korrektheit der in der Applikation erkannten Transportmittel. Dabei geht es um die von dem Benutzer in Echtzeit verwendeten Transportmittel. Bei einem **Mittelwert** von **5,20** liegt die Bewertung dabei im oberen Bereich und ist gleichzusetzen mit der Aussage „stimme eher zu“. Aus **Abbildung 64** lässt sich ableiten, dass die Transportmittelerkennung bei 50 von 64 Personen (Summe über die positiven Kategorien (5), (6) und (7)) funktioniert hat, bei sechs Personen hat die Funktionsweise keinen nennenswerten Eindruck hinterlassen und

lediglich bei 8 von 64 Personen ($\approx 12,5\%$) hat die Transportmittelerkennung versagt. Es lässt sich dabei eine klare Häufung bei den Bewertungen um die Kategorie (5 – stimme eher zu) und (6 – stimme zu) feststellen.

"Die verwendeten Transportmittel wurden von dem intNAV korrekt erkannt."

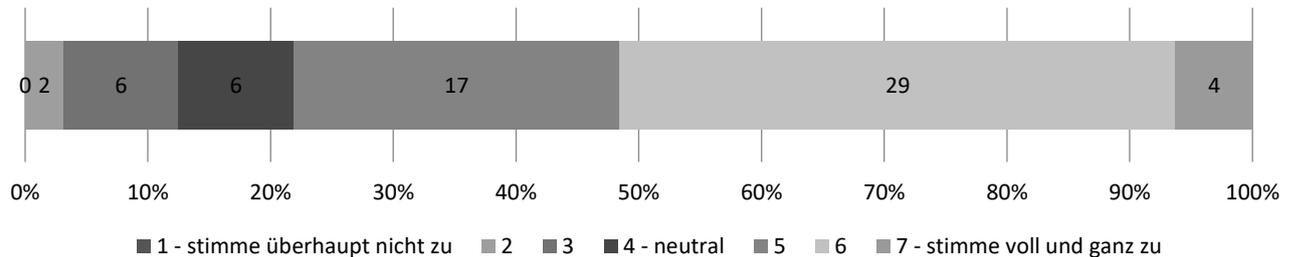


Abbildung 64: Verteilung der Ergebnisse zur Transportmittelerkennung in Echtzeit

Erkennung von aktiven Anomalien

Mit der nächsten Frage sollten die Studienteilnehmer die Korrektheit der aktiven Anomalieerkennung bewerten. Da die Erkennung von passiven Anomalien keinen neuartigen Charakter hat und in vergleichbaren Applikation zu dem vorliegenden Thema schon umgesetzt ist, interessiert in der nachfolgenden Auswertung die Erkennung von aktiven Anomalien. Diese ist, zum aktuellen Zeitpunkt, in dem hier vorgestellten Umfang noch nicht in vergleichbaren Applikationen umgesetzt. Wie aus **Abbildung 65** hervorgeht, liegt der **Mittelwert** der Bewertungen bei **4,28** und dabei im neutralen Bereich. Jedoch lässt sich eine positive Tendenz vernehmen, mit 26 Probanden welche die Frage positiv bewertet haben im Vergleich zu nur 14 Probanden, welche diese Frage negativ bewertet haben. Es gibt auch Ausreißer in beiden Bewertungskategorien mit 2 Probanden welche diese Frage jeweils mit der höchsten Bewertung von (7 – stimme voll und ganz zu) und mit der niedrigsten Bewertung (1 – stimme überhaupt nicht zu) assoziiert haben.

"Die aufgetreten aktiven Abweichungen wurden von dem intNAV korrekt erkannt."

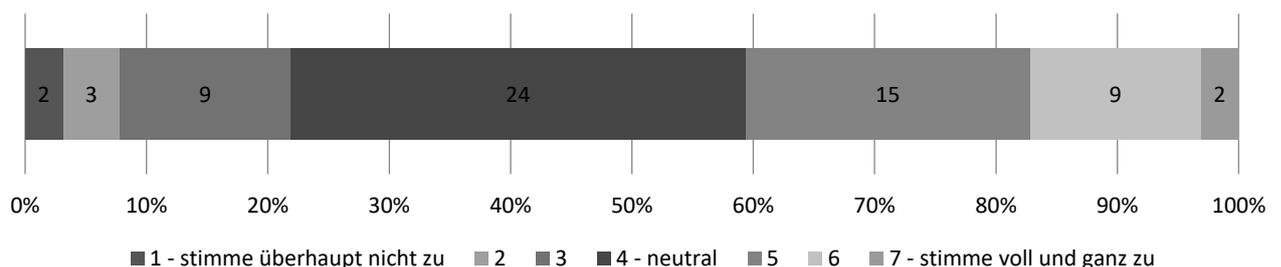


Abbildung 65: Verteilung der Ergebnisse zur Anomalieerkennung

Performanz der Applikation

Die letzte Frage zur Funktionalität der Applikation beschäftigt sich mit der Performanz. Die Probanden sollten dabei bewerten, ob es sich bei den Daten, welche in der Anwendung verwendet wurden, um Echtzeitdaten handelt. Dies ist eine mögliche Art, die Performanz zu testen, da beispielsweise Latenzzeiten und Akkuverbrauch

von den Probanden schwer zu testen sind. Wie aus **Abbildung 66** hervorgeht, schneidet die entwickelte Applikation mit dem in Kapitel 5 vorgestellten Ansatz zum Umgang mit Echtzeitdaten überdurchschnittlich gut ab. Mit einem **Mittelwert** von **5,83** liegt die Bewertung dabei fast bei (6 – stimme zu). Dies ist gleichzeitig beste Bewertung von allen Messgrößen. Es fällt ebenfalls in **Abbildung 66** auf, dass überdurchschnittliche viele Probanden, die Applikation mit der Note (7 – stimme voll und ganz zu) bewerten. In Summe bewerteten 60 von 64 Probanden (≈93,75%) diesen Punkt als positiv.

"Die Anwendung hat stets mit den aktuellsten Daten in Echtzeit gearbeitet."

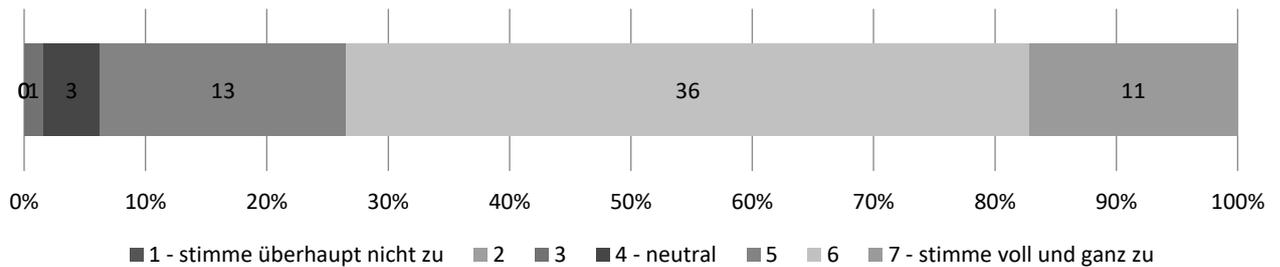


Abbildung 66: Verteilung der Ergebnisse zur Performanz der Applikation

Kommunikation von Störungen

In der Umfrage (vgl. Kapitel 6.1.4) wurden Arten von Unterbrechungen untersucht, welche die Benutzer bei aktiven und passiven Anomalien bevorzugen. Dabei kam sowohl für die aktiven als auch für die passiven Anomalien das gleiche Ergebnis heraus, nämlich, dass den Benutzern eine einmalige Kommunikation der vorliegenden Störung unter Angabe der Klassifizierung, Alternativen und Konsequenzen am liebsten ist. Das Resultat dieser Umfrage wurde in den Prototypen umgesetzt und die Erfolgsquote mit Hilfe der Benutzerstudie untersucht. Mit einem **Mittelwert** von **4,28** (vgl. **Abbildung 67**) liegt die Bewertung dabei im mittleren Bereich mit einer leichten Tendenz ins Positive, denn 28 Personen bewerteten diese Frage eher positiv und $\frac{1}{4}$ der Probanden bewertet diese negativ.

"Erkannte Störungen wurden auf eine angemessene Art und Weise kommuniziert."

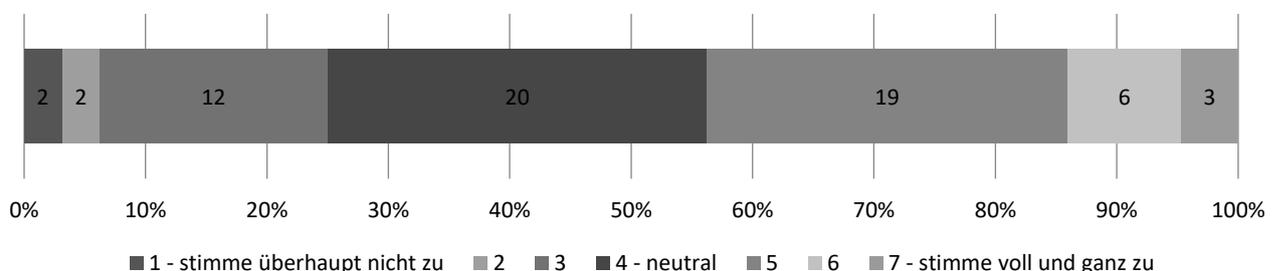


Abbildung 67: Verteilung der Ergebnisse zur Kommunikation von Störungen

Empfehlungen

In der Applikation wurden die Benutzer mit einer Auswahl an Empfehlungen konfrontiert. Dazu zählt die Empfehlung zum Losgehen, die Empfehlung, dass sich der Benutzer beeilen soll, wenn dieser die geplante Route einhalten möchte, die Empfehlung lt. Routenplan umzusteigen und die Empfehlung besser an einem anderen Punkt in der Nähe einzusteigen. Die Korrektheit dieser vier Empfehlungen sollte, falls die Benutzer diese erhalten haben, bewertet werden. Unter Korrektheit werden der Zeitpunkt und die jeweils richtige Information verstanden. Der **Mittelwert** von **4,47** aus **Abbildung 68** entspricht eher der neutralen Bewertung, jedoch ist auch hier eine positive Tendenz zu vermelden mit 35 Probanden, welche diese Frage positiv beantwortet haben und 17 Probanden, welche die Frage negativ beantwortet haben.

"Die angezeigten Empfehlungen bei erkannten aktiven Störungen waren hilfreich."

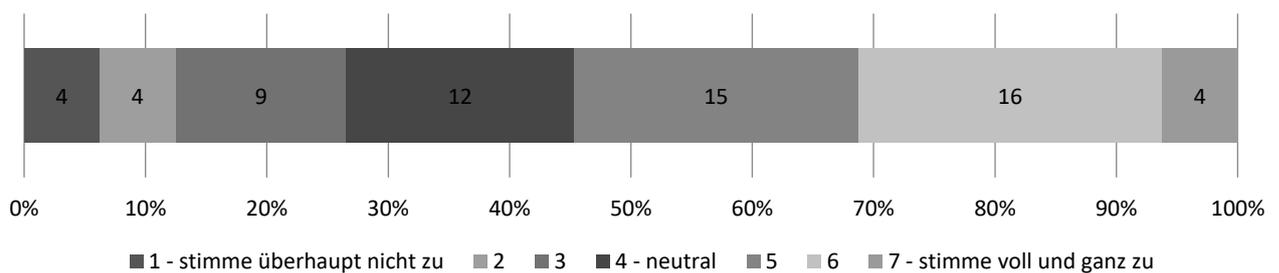


Abbildung 68: Verteilung der Ergebnisse zu den gezeigten Empfehlungen

Informationsdichte

Unter Informationsdichte wird, in dem hier getesteten Zusammenhang, die zur Verfügung stehende Information zur Route auf dem Display eines Smartphones verstanden. Im Rahmen dieser Frage ist von Interesse, in wieweit die Probanden die Information, welche ihnen präsentiert wurde, als ausreichend empfanden oder ob ihnen Information gefehlt hatte. Wie das Ergebnis **Abbildung 69** vermuten lässt, mit einem **Mittelwert** von **3,45**, haben die Probanden im Rahmen der Ausführung des Prototypen Informationen vermisst, welche ihnen weitergeholfen hätte. Die Tendenz ist in diesem Bereich eher negativ mit 35 Probanden und mit nur 13 Probanden positiv zu sehen. Auch das Fehlen der besten Bewertung (7 – stimme voll und ganz zu) mit gleichzeitig 3 sehr schlechten Bewertungen (1 – stimme überhaupt nicht zu) bestätigt das vorliegende Meinungsbild.

"Die dargestellten Informationen innerhalb der Applikation beurteile ich als ausreichend."

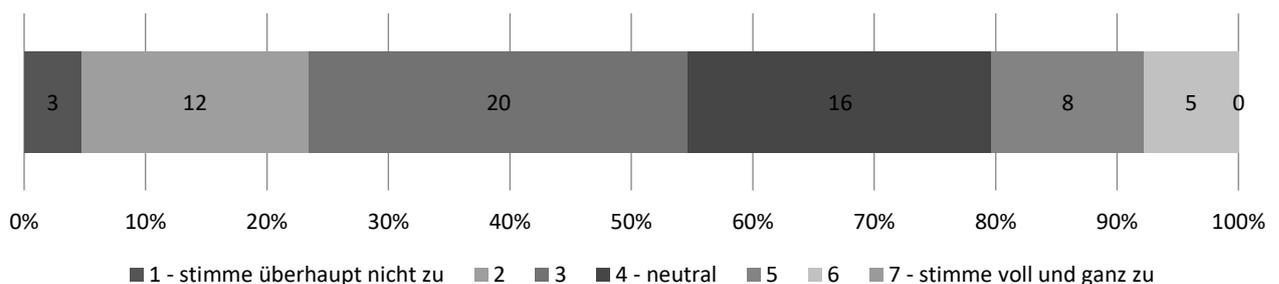


Abbildung 69: Verteilung der Ergebnisse zur Informationsdichte

Wiederkehrwahrscheinlichkeit

Abschließend sollte die Frage geklärt werden, in wie weit die Benutzer mit der Anwendung so zufrieden waren, dass sie diese wiederverwenden würden oder sie weiterempfehlen würden. Das Ergebnis der Fragebögen zeigt eine eindeutige positive Tendenz mit einem **Mittelwert** von **5,42**. Wie in **Abbildung 70** zu sehen ist, bewerten 52 Personen diese Frage mit positiv und nur drei Personen mit negativ. Positiv ist auch anzumerken, dass von allen getesteten Personen insgesamt neun die höchste Bewertung abgegeben haben und keine einzige Testperson die schlechteste Bewertung abgegeben hat.

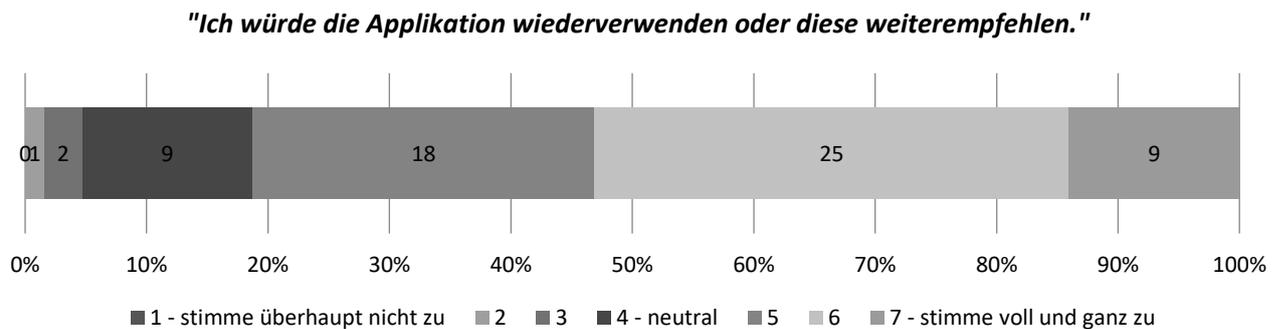


Abbildung 70: Verteilung der Ergebnisse zur Wiederkehrwahrscheinlichkeit

Die Interpretation der Ergebnisse kann im Kapitel 6.3 nachvollzogen werden. Im nächsten Abschnitt sollen die Ergebnisse aus den Abschlussinterviews berichtet werden.

ABSCHLUSSINTERVIEW

Im Gegensatz zu den Tagebüchern, wurden die Abschlussinterviews einmalig durchgeführt. Das Ziel dieser Interviews war es zusätzlich zu der quantifizierten Befragung in den Tagebüchern qualifiziertes Feedback zur Applikation zu erhalten. Der Fokus bei dem Abschlussinterview war es herauszufinden, welche Punkte den Benutzern positiv und welche negativ in Erinnerung geblieben sind. Das Interview wurde so gestaltet, dass nicht jeder Punkt beantwortet werden muss.

Nachfolgend sind sämtliche Fragen aus dem Abschlussinterview mit den Ergebnissen dargestellt.

Zunächst sollen die Situationen ermittelt werden, in denen das intNAV von den Probanden verwendet wurde.

Wenig überraschend ist die Auswertung der ersten Frage. Da alle Testpersonen Studenten (Bachelor und Master) an der LMU sind, gaben 64 Personen die Fahrt zur und von der LMU als Verwendungszweck des intNAVs an und 28 Personen verwendeten die Anwendung um in die Arbeit zu fahren (vgl. **Abbildung 71**). Dies ist umso interessanter, als es sich bei diesen beiden Zielen um Strecken handelt, welche die Benutzer (fast) täglich fahren. Somit ist anzunehmen, dass die Navigation dabei nicht im Vordergrund stand. Fast genauso häufig wurde es für die Fahrt in die Innenstadt verwendet. Bei diesem Punkt bleibt offen, ob das Ziel „Innenstadt“ auch das Ziel „Lokal“ abdeckt. Einen Zweck für die Fahrt in die Innenstadt wurde von den Testpersonen nicht angegeben.

„Bitte lassen Sie die vergangene Woche nochmals Revue passieren – in welchen Situationen haben Sie das intNAV verwendet?“

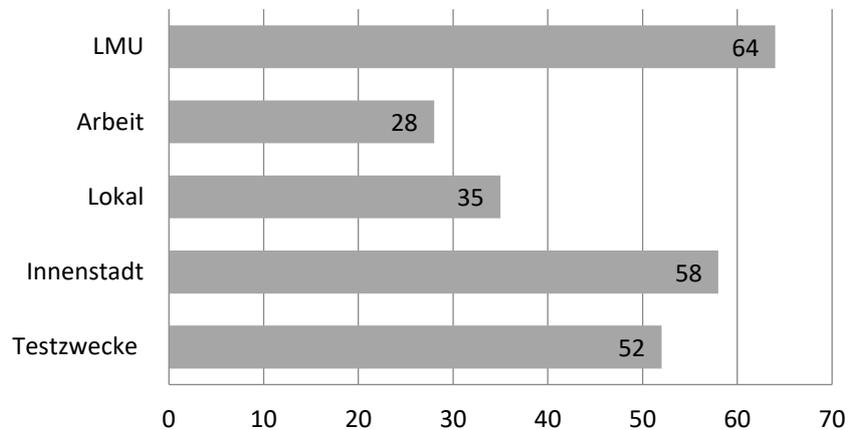


Abbildung 71: Überblick über die Situationen, in welchen das intNAV von den Probanden verwendet wurde

Die meisten Nutzer haben die Applikation absichtlich getestet, indem sie nicht die von dem intNAV vorgegebene Route ausgewählt haben, sondern absichtlich in eine falsches Transportmittel eingestiegen sind oder eine Station früher ausgestiegen sind. Von vornherein ausgeschlossen wurden dabei Langstreckenfahrten, da der hier entwickelte Prototyp zunächst nur in München funktioniert.

Bei den nächsten beiden Fragen im Abschlussinterview werden Situationen abgefragt, in denen die Benutzer die Anwendung als besonders hilfreich und als überflüssig empfanden.

In welchen Situationen war die Applikation am hilfreichsten³⁰?

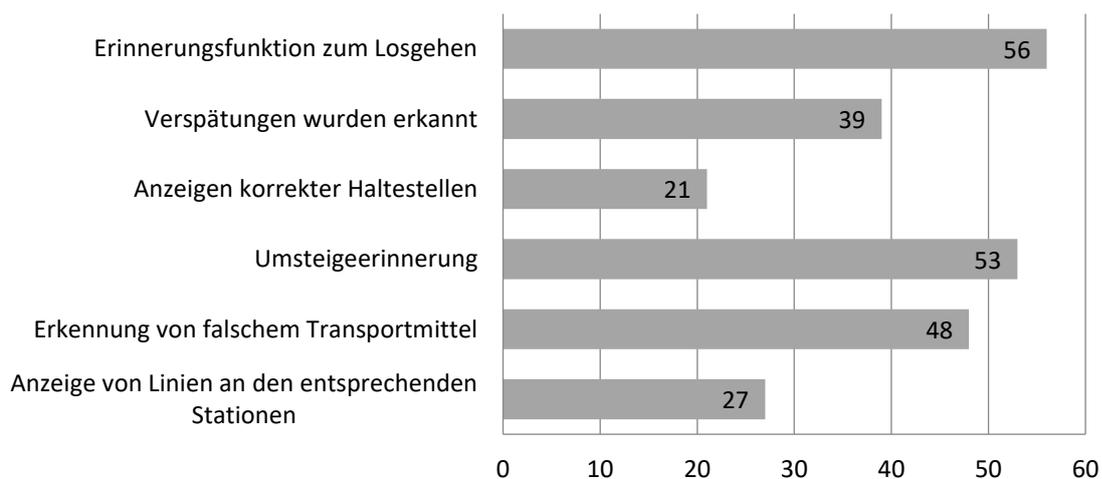


Abbildung 72: Überblick über Situationen in denen die Applikation hilfreich war

³⁰ Ähnliche Antworten wurden zusammengefasst

Wie in **Abbildung 72** zu sehen ist, erschien den meisten Testpersonen die Erinnerungsfunktion (zum Losgehen und zum Umsteigen) am hilfreichsten. Wohingegen die im Rahmen dieser Arbeit entwickelte Funktionen, Transportmittelerkennung, vergleichsweise als hilfreich aber doch unwichtiger als die Erinnerungsfunktionen erschienen. Dies hat aber einen einfachen Hintergrund. Da die Testpersonen die Anwendung, wie in der Frage zuvor, auf ihren gewohnten Routen verwendet haben, waren sie über die Transportmittel, welche sie verwenden sollten, bestens informiert. Somit wurde diese Funktion im Vergleich seltener verwendet als die Erinnerungsfunktion. Im nachfolgenden Kapitel 6.3 dieses Ergebnis nochmal diskutiert.

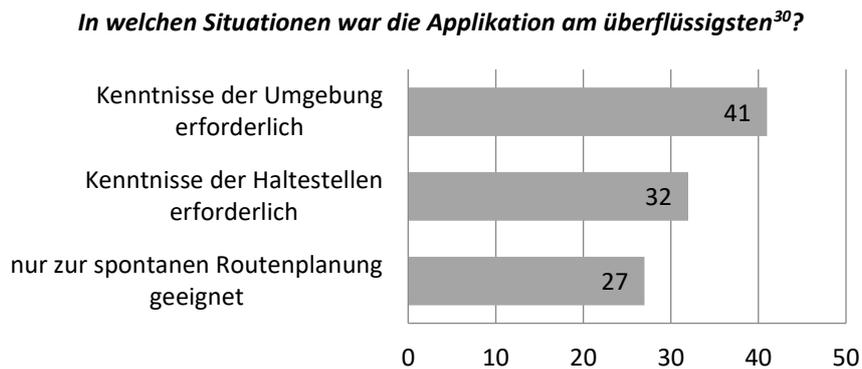


Abbildung 73: Überblick über Situationen in denen die Applikation überflüssig erschien

Überflüssig wurde die Anwendung in Situationen erachtet, in welcher der Benutzer sich in der entsprechenden Umgebung selbst sehr gut auskennen sollte (vgl. **Abbildung 73**). Dies liegt daran, dass in der Anwendung keine Karte zur Fußnavigation implementiert wurde, da dies nicht der Fokus der vorliegenden Arbeit ist. Weiterhin ist es richtig, dass die Anwendung, sobald eine entsprechende Linie erkannt wurde, die Haltestellen anzeigt. Natürlich ist es nur dann hilfreich, wenn man sich in der Gegend auskennt und etwas mit den Haltestellen anzufangen weiß. Auch dieser Punkt wurde, bei der Entwicklung des Prototypen, vorausgesetzt und war nicht im Fokus der Entwicklung. Die Anwendung ist auch so konzipiert, dass nur unmittelbar bevorstehende Routen von dem intNAV erkannt und „begleitet“ werden können. Diese Funktion kann in weiterführenden Arbeiten zu diesem Thema weiter untersucht werden.

Anschließend wurden die Probanden gebeten, drei Punkte aufzuzählen, welche ihnen positiv und welche ihnen negativ im Gedächtnis geblieben sind.

Nennen Sie 3 Punkte, die Ihnen positiv im Gedächtnis geblieben sind³¹.

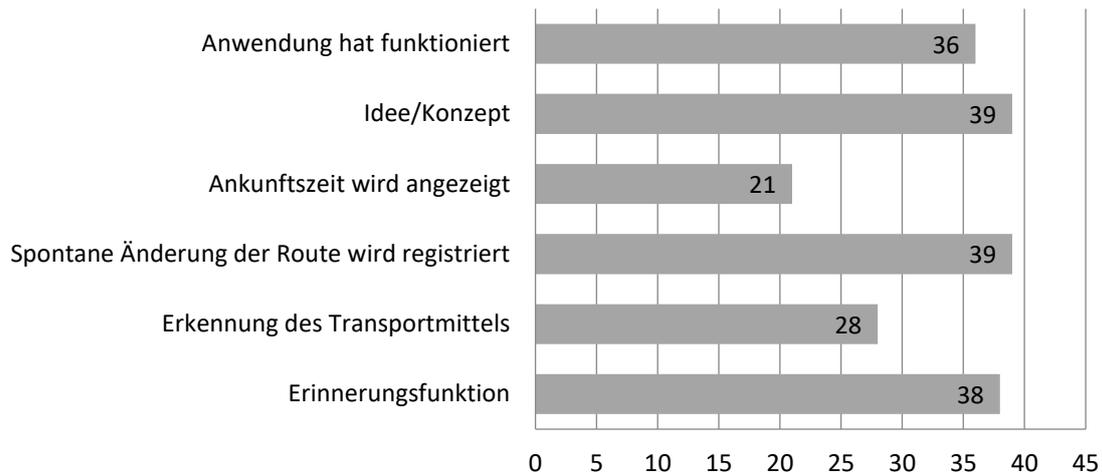


Abbildung 74: Punkte, welche den Probanden positiv im Gedächtnis geblieben sind

Nennen Sie 3 Punkte, die Ihnen negativ im Gedächtnis geblieben sind³¹.

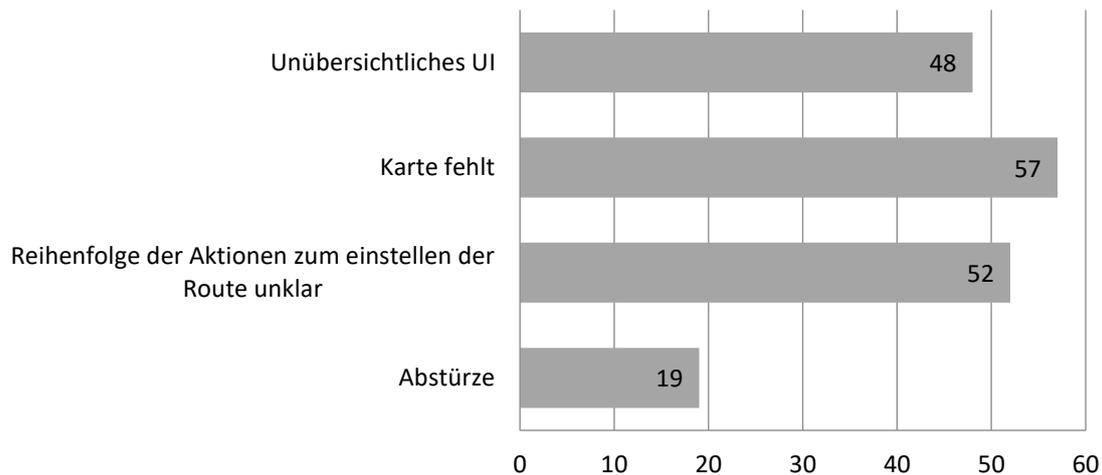


Abbildung 75: Punkte, welche den Probanden negativ im Gedächtnis geblieben sind

Nicht verwunderlich ist die Aufzählung der Punkte, welche nicht überzeugt haben. Das UI, das Fehlen der Karte oder die Reihenfolge der Aktionen zum Einstellen der Route sind alles Punkte, welche bei der Entwicklung des hier vorliegenden Prototyps nicht im Vordergrund standen. Diese Punkte, werden jedoch bei aktuellen Anwendungen vorausgesetzt, sodass die Haltung auch dieser Anwendung gegenüber die gleiche gewesen ist. Da der Fokus dieser Arbeit bei der Konzeptionierung eines intNAVs liegt, freut es umso mehr, dass die Hauptfunktionalitäten (Transportmittelerkennung und Anomalieerkennung), wie in **Abbildung 74** zu sehen ist, als positiv im Gedächtnis der Benutzer geblieben sind. UI Anpassungen, Reihenfolge bei der Routenwahl oder das Fehlen der Karte (vgl. **Abbildung 75**) sind zwar valide Punkte, sind aber nicht im Fokus der vorliegenden Arbeit gewesen, bei der es um die Funktionalität ging. Vermutlich wurde dieser Punkt im Eingangsgespräch mit

³¹ Ähnliche Antworten wurden zusammengefasst

den Personen nicht stark betont. Es freut umso mehr, dass bei den negativen Punkten keine Hauptfunktionalitäten auftauchen, sondern nur der Bedienung der Anwendung geschuldete Probleme.

Im Verlauf der vorliegenden Arbeit wurde stets der Vergleich zu traditionellen Navigationssystemen gezogen. Da liegt es nahe, auch diesen Vergleich an die Testpersonen weiterzugeben. Deshalb beschäftigte sich die vorletzte Frage im Abschlussinterview damit, herauszufinden, welche Funktionalitäten die Personen im Vergleich zu traditionellen Navigationssystemen vermissen.

Was haben Sie an der Applikation am meisten vermisst (im Vergleich mit anderen Navigationssystemen)³²?

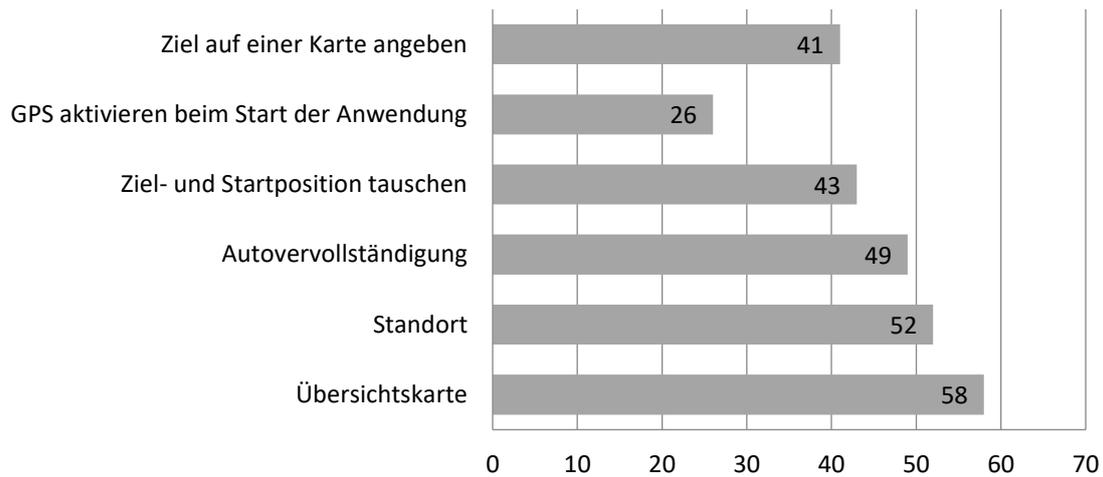


Abbildung 76: Vergleich der Funktionen eines intNAVs mit einem traditionellen Navigationssystem

Erfreulicherweise handelt es sich bei den Funktionen, welche im Vergleich zu traditionellen Navigationssystemen im intNAV vermisst werden könnten, zum einen um Funktionen welche schnell implementiert werden können und zum anderen um Funktionen, welche auch nicht Teil der vorliegenden Arbeit sind. Somit ist die Funktionalität des intNAVs auch hier nicht bemängelt worden.

Schlussendlich sollte nochmal die Frage geklärt werden, ob die Erwartungen an das intNAV enttäuscht, erfüllt oder übertroffen wurden. **Abbildung 77** zeigt dabei eindrucksvoll, dass bei genau $\frac{3}{4}$ der Befragten, die Erwartungen an das intNAV erfüllt wurden und bei 8% sogar übertroffen. Lediglich bei 17% der Probanden wurden die Erwartungen an das intNAV enttäuscht.

³² Ähnliche Antworten wurden zusammengefasst

Wurden Ihre Erwartungen an die Applikation enttäuscht, erfüllt oder übertroffen?

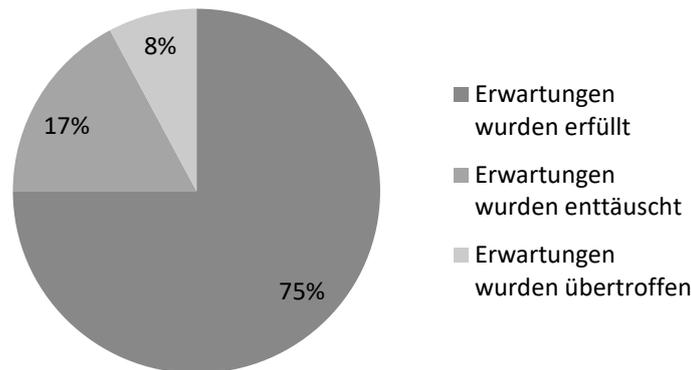


Abbildung 77: Verteilung der Erwartungserfüllung

6.3 INTERPRETATION DER ERGEBNISSE

In Summe waren die Ergebnisse sowohl der Tagebücher als auch der Abschlussinterviews sehr zufriedenstellend. Die Tagebücher erlaubten eine quantifizierbare Gegenüberstellung der Ergebnisse wohingegen die Abschlussinterviews einen Eindruck über die allgemeine Zufriedenheit der Anwendung verschaffen sollte. Was die Funktionalität der Applikation angeht, wurde diese durchweg positiv bewertet. Die **Transportmittelerkennung** und **Anomalienerkennung** lag dabei im oberen positiven Bereich und die mit einer zusätzlichen Tendenz zum Positiven. Die Anzahl der negativen Bewertungen kann dabei beinahe vernachlässigt werden. Besonders herausgestochen hat die **Performanz** der Applikation mit beinahe ausschließlich positiven Bewertungen. Dies spricht für den Einsatz der verwendeten **CEP**-Architektur. Die Vorteile kommen, wie in Kapitel 5 beschrieben, insbesondere bei mehr als einem Anwender zu tragen. Um die Akzeptanz der Anwendung zu testen wurden dabei die Punkte **Kommunikation** von Störungen, **Empfehlungen**, **Informationsdichte** sowie **Wiederkehrwahrscheinlichkeit** abgefragt. Bis auf den Punkt Informationsdichte, waren die Auswertungsergebnisse der übrigen Punkt zufriedenstellend. Alle bewegten sich im neutralen bis positiven Bereich.

7 ZUSAMMENFASSUNG

In der vorliegenden Arbeit wurde das Konzept eines intermodalen Navigationssystems vorgestellt. Nachfolgend werden die Ergebnisse der einzelnen Kapitel dieser Arbeit zusammengefasst (1). Anschließend werden der entwickelte Prototyp sowie die Studienergebnisse diskutiert (2). Dabei werden interessante Beobachtungen hervorgehoben und als wissenschaftlicher Beitrag dieser Arbeit zusammengefasst. Im letzten Abschnitt (3) werden Ideen vorgestellt, inwiefern der hier vorgestellte Ansatz für intNAVs weiterentwickelt werden kann. Der Fokus liegt dabei auf einem sparsameren Umgang mit den Ressourcen eines Smartphones, auf der Implementierung weiterer, insbesondere unterirdisch fahrender Verkehrsmittel sowie in der Betrachtung einer verbesserten Benutzeroberfläche.

7.1 ZUSAMMENFASSUNG DER KAPITEL

Das **Kapitel 1** diente der allgemeinen Einordnung des Problems und der Identifizierung einer *Forschungslücke* im Bereich intermodaler Navigationssysteme. Dabei wurden aktuelle Entwicklungen in Bezug auf das Transportverhalten der Menschen in Städten analysiert und daraus das Fehlen eines Äquivalents zum traditionellen Navigationssystem für den intermodalen Gebrauch abgeleitet. Im ersten Kapitel wurde somit die Notwendigkeit der Entwicklung eines intNAVs aufgezeigt.

Bevor jedoch die notwendigen Schritte für die Entwicklung eines intNAVs im weiteren Verlauf der vorliegenden Arbeit diskutiert werden konnten, wurden in **Kapitel 2** die grundlegenden Terminalien sowie der aktuelle Stand der Forschung vorgestellt. Dies dient zum einen dazu, Begrifflichkeiten klar voneinander abzutrennen und einen Standard zu etablieren, welcher sich durch die gesamte vorliegende Arbeit durchzieht. Weiterhin wird dadurch eine bessere Lesbarkeit erreicht. Zum anderen soll dieses Kapitel klar die bestehende Literatur zum Thema intermodale Navigationssysteme aufzeigen und die Forschungslücke, welche behandelt werden soll, begründen. Das zweite Kapitel zeigt zudem den Aufbau und die Fundamente der vorliegenden Dissertation auf.

Eine der Fundamente der vorliegenden Dissertation - die intermodale Transportmittelerkennung - wird in **Kapitel 3** observiert. Dabei wird die entwickelte Funktionsweise anhand von Zustandsautomaten sowie die Erkennungswahrscheinlichkeit der einzelnen Zustände anhand eines durchgeführten Experiments offenbart. Der vorgestellte Ansatz bedient sich dem Konstrukt des logischen Schließens bei der Erkennung der

Wahrscheinlichkeit der Zustände und bedient sich weiterhin der verfügbaren Sensoren eines Mobiltelefons. Bei dem durchgeführten Experiment erreicht die Erkennungswahrscheinlichkeit einen Wert von knapp 96%, welcher vergleichbare Ansätze aus der Literatur übertrifft.

Das zweite Fundament bildet die Erkennung der *Wahrscheinlichkeit* einer vorliegenden Abweichung in Echtzeit. Dabei soll ermittelt werden, wie hoch die Wahrscheinlichkeit einer aktiven Abweichung ist, denn die Erkennung von passiven Abweichungen ist trivial. Das Ziel dieser Abweichenerkennung liegt in einer smarten Kommunikation mit dem Benutzer, um diesen darauf hinzuweisen, ob und – wenn ja – wie dieser das gewünschte Verkehrsmittel erreichen kann, wenn es zu einer aktiven Abweichung kam. Das **Kapitel 4** widmet sich dieser Thematik. Eine korrekte Erkennung einer Abweichung stellt aus dem Grund eine fundamentale Funktionsweise eines intNAVs dar, da es die Differenzierung zwischen einer bereits existierenden intermodalen *Routenplanung* und einer in dieser Arbeit entwickelten intermodalen *Routennavigation* ermöglicht. Die vorgestellte Lösung zur Erkennung der Wahrscheinlichkeit einer Abweichung in Echtzeit bedient sich dem Konstrukt der Neuronalen Netze. Der vorgestellte Ansatz berechnet basierend auf der Abweichung eine bessere Route. Dabei stellt der vorgestellte Ansatz zur Verwendung von Neuronalen Netzen zur Erkennung von Abweichungen ein Novum dar. Zusammen mit dem Ergebnis aus Kapitel 3 werden somit die *Zustandserkennung* und die *Abweichenerkennung* für intermodale Navigation erstmals ermöglicht.

In **Kapitel 5** wird die Anwendbarkeit eines optimierten Vorgehens (CEP) zum schnelleren Durchsatz von sehr großen Datenmengen in Echtzeit auf das entwickelte intNAV untersucht. In Kapitel 5 wurde gezeigt, dass der entwickelte Prototyp zwar eine Anwendbarkeit von CEP zulässt, jedoch in dem hier vorgestellten Kontext seine volle Performanz nicht vollständig zeigen konnte. Jedoch konnte mit einem Lasttest gezeigt werden, dass CEP durchaus eine Lösung darstellt, um das hier entwickelte intNAV in einer Großstadt wie beispielsweise München betreiben zu können. Dabei wäre es in der Lage mit 250.000 Anfragen pro Sekunde in Echtzeit fertig zu werden.

Um den in Kapitel 5 entwickelten Prototypen hinsichtlich Funktionsweise und Nutzerakzeptanz zu testen, wurden in **Kapitel 6** eine Umfrage sowie eine qualitative Studie durchgeführt. Das Ziel der Umfrage war es zum einen die *Notwendigkeit* eines intNAVs zu erfragen und zum anderen *Kriterien* zu identifizieren, welche den Benutzern besonders wichtig bei der Verwendung des intNAVs auf dem Smartphone erscheinen. Das Ergebnis bzgl. der Notwendigkeit war eindeutig. Bei 83% der Befragten wurden die Erwartungen an das intNAV entweder erfüllt oder übertroffen. Die ermittelten Kriterien aus der Umfrage wurden daraufhin in der qualitativen Studie mit 64 Teilnehmern untersucht. Die Kriterien „Performanz der Applikation“, „Wiederkehrwahrscheinlichkeit“ und „Echtzeit- und Anomalienerkennung“ wurden erfüllt, wohingegen das Kriterium „Informationsdichte“ noch weiterer Verbesserung bedarf.

7.2 DISKUSSION

Nach eingehender Literaturrecherche (siehe Kapitel 2) wurde eine Forschungslücke erkannt, welche die Erstellung eines Konzepts für intermodale Navigationssysteme in den Vordergrund stellt. Da kein vergleichbares Konzept existiert, reichte es im Rahmen der vorliegenden Arbeit aus, ein funktionierendes Proof-of-Concept (PoC) zu erstellen. Der Vorteil bei einem PoC ist, dass dieses lediglich die Funktionsweise untersuchen soll,

sämtliche Optimierungen werden hierbei vernachlässigt und sind Gegenstand weiterführender Versionen (siehe 7.3). Obwohl die Funktionsweise des im Rahmen der vorliegenden Dissertation entwickelten intNAVs erfolgreich getestet wurde (siehe Kapitel 6), wäre ein marktreifer, auf Gewinn ausgerichteter Einsatz dieses Prototypen nicht zu empfehlen, da bei der Umsetzung des PoC die Performanz nicht im Vordergrund stand, sondern lediglich der Beweis, dass eine Implementierung des vorgestellten Ansatzes – wie in Kapitel 3 diskutiert – funktioniert. Ob dabei der Akku eines Mobiltelefons sehr schnell leer wird, da die Anwendung in regelmäßigen, redundanten Abständen die Geoposition bezieht, wurde bei den Tests vernachlässigt. Es ging lediglich darum, eine Lösung aufzuzeigen, wie intermodale Zustände zuverlässig erkannt werden können. Neben der Performanz liegt ein weiterer Nachteil der vorliegenden Lösung für die Transportmittelerkennung, wie sie bei der vorliegenden Arbeit umgesetzt wurde, darin, dass diese auf die Aktualität der Daten angewiesen ist. Diese wurden im Rahmen des PoC manuell ermittelt und werden nicht automatisch bei jeder Aktualisierung von Key-Points gepflegt. Um eine gewerbliche Verwendung des vorgestellten intNAVs zu ermöglichen, müssten Entwickler die zu Grunde liegende Datenbasis automatisiert aktuell halten. Diese Aspekte der *Performanz* sowie *Aktualität der Daten* wurden im Rahmen der vorliegenden Arbeit vernachlässigt.

Ähnlich wie klassische Navigationssysteme, soll auch ein intNAV in einer Großstadt von sehr vielen Menschen genutzt werden. Um den entwickelten Prototyp auf diese Last vorzubereiten, wurde CEP zur Optimierung des Datendurchsatzes ausprobiert. Die Anwendung von CEP in diesem Kontext ist in der Literatur ein Novum und wurde im Rahmen der vorliegenden Arbeit erfolgreich gezeigt. Jedoch konnte intNAV mit dem zugrundeliegenden CEP-Konzept nicht ausreichend getestet werden, da es unmöglich war, an die gewünschte Nutzerzahl von ca. 700.000 Personen zu gelangen, welche theoretisch intNAV in München verwenden würden (vgl. Kapitel 5). Aus diesem Grund wurde die Performanz von CEP im intNAV mittels einer Simulation getestet. Das Ergebnis war zufriedenstellend. Dieser Last-Test wurde im Rahmen der vorliegenden Arbeit simuliert und basiert nicht auf echten Daten. Von Interesse wäre nun ein Test des intNAVs mit CEP unter realen Bedingungen. Gegebenenfalls wurden bei der Simulation Aspekte nicht berücksichtigt, welche in Wirklichkeit jedoch von Relevanz sein könnten und das ermittelte Ergebnis von 205.000 Anfragen pro Sekunde verfälschen.

Die in Kapitel 6 durchgeführte Umfrage bzgl. der Notwendigkeit und den Kriterien eines intNAVs war sehr zufriedenstellend. Basierend auf den Ergebnissen der Umfrage wurde der Prototyp entwickelt. Die Ergebnisse der anschließend durchgeführten Studie mit 64 Teilnehmern geben jedoch nur Rückschlüsse auf eine begrenzte Zielgruppe (Alter zwischen 19 und 26 mit hauptsächlich Studenten). Um repräsentative Ergebnisse zu erhalten, müssten weitere Studien durchgeführt werden, welche nicht nur zum einen eine größere Teilnehmerzahl ansprechen, sondern ebenfalls ein breiteres Teilnehmerfeld haben. Weiterhin sollen die Ergebnisse der durchgeführten Studie als Input für weitere Studien herangezogen werden. Im Rahmen der vorliegenden Arbeit waren eine größere Teilnehmerzahl und eine breitere Streuung der Teilnehmer aufgrund der Zeitkritikalität nicht möglich.

Trotz der genannten Limitationen sind die erreichten Ergebnisse des PoC sehr vielversprechend. Nicht nur wurden bei dem im Rahmen der hier vorliegenden Arbeit erstellten Prototypen bessere Resultate bei der Transportmittelerkennung (verglichen mit der durchgeführten Literaturanalyse im Kapitel 2) sowie der

Abweichenerkennung erzielt, es wurde auch erstmals ein funktionierendes Konzept für ein intNAV vorgestellt. Zum ersten Mal kann der Benutzer nun in Echtzeit über mehrere Transportmittel innerhalb einer Route hinweg navigiert werden.

Im Rahmen der vorliegenden Arbeit wurden erstmalig Kriterien an das intNAV ermittelt, welche im Vorfeld in einer Umfrage erfasst wurden. Dabei handelt es sich um funktionale und nicht-funktionale Anforderungen an das intNAV. Diese Kriterien bilden einen Leitfaden für weiterführende Arbeiten mit intNAVs.

Zusammengefasst sind nachfolgend die Hauptbeiträge der vorliegenden Dissertation aufgeführt:

Forschungsbeitrag 1: Eine **umfassende und detaillierte Untersuchung** der existierenden Literatur zum Thema *intermodale Navigation* mittels Sensoren von Mobiltelefonen wurde durchgeführt und liegt vor. Dabei wurden zunächst Sensoren von Mobiltelefonen, welche der Navigation dienlich sein könnten, aufgezeigt und anschließend die Erkennungswahrscheinlichkeit in Bezug auf die gängigsten intermodalen Zustände („zu Fuß“, „Fahrrad“, „Auto“, „U-Bahn“, „S-Bahn“, „Tram“, „Zug“, „Bus“) der Literatur entnommen. **Tabelle 1** (Seite 29.) zeigt einen Überblick über die Ergebnisse der durchgeführten Literaturrecherche in Bezug auf die Eignung der Mobiltelefonsensoren zur intermodalen Navigation.

Forschungsbeitrag 2: **Notwendigkeit** sowie **Kriterien** eines intNAVs wurden repräsentativ erhoben. Da ein vergleichbares Konzept eines intNAVs in der Literatur nicht existiert, wurde zunächst die Notwendigkeit in einer repräsentativen Umfrage nachgewiesen. **Tabelle 15** (S. 134 f.) zeigt Kriterien für intNAVs, welche aus dieser Umfrage abgeleitet wurden, auf. Diese sind nicht nur für die vorliegende Arbeit, sondern auch für weiterführende Arbeiten mit intNAVs von sehr starkem Interesse.

Forschungsbeitrag 3: Funktionierender **intNAV-Prototyp** mit einer vielversprechenden Zustandserkennung sowie Abweichenerkennung liegt vor. Erstmals liegt ein funktionierender Prototyp vor, welcher in der Lage ist, einen Benutzer über mehrere Transportmittel innerhalb einer Route hinweg in Echtzeit zu navigieren. Im Rahmen der vorliegenden Arbeit wurde dabei ein Beitrag sowohl zur intermodalen Zustandserkennung (Kapitel 3) als auch zur Abweichenerkennung (Kapitel 4) geleistet. Die erzielten Ergebnisse übertreffen dabei die Ergebnisse aus der Literatur. Die Probanden in der durchgeführten Studie schätzen dabei die schnelle und akkurate Transportmittel- und Abweichenerkennung (**Tabelle 20**, S. 145)

Forschungsbeitrag 4: Erfolgreiche **Anwendbarkeit von CEP** innerhalb einer neuen Domain. Das Haupteinsatzgebiet von CEP ist der Aktienhandel. Die Betrachtung von CEP für das vorliegende Thema ist neu, jedoch auch naheliegend. Im Rahmen der vorliegenden Arbeit wurde CEP erfolgreich für das intNAV eingesetzt (Kapitel 5).

7.3 AUSBLICK

Basierend auf dem in dieser Arbeit entwickelten intNAV-Konzept lassen sich Vorschläge für weiterführende Arbeiten ableiten.

Wie im Einführungskapitel beschrieben und in der Umfrage bestätigt wurde, eignen sich intNAVs ausschließlich für mobile Endgeräte; am weitesten verbreitet ist dabei das Mobiltelefon. Die Ressourcen des Mobiltelefons sind jedoch endlich und müssen optimal ausgenutzt werden, da sonst die Anwendung von den Benutzern schnell verworfen wird. Der hier vorgestellte Ansatz vernachlässigte die Ressourceneinsparung, da in erster Linie ein State-of-the-Art-Ansatz im Bereich intNAVs entwickelt werden soll. Damit das hier vorgestellte Konzept basierend auf dem Key-Point-Ansatz jedoch erfolgreich und flächendeckend verbreitet werden kann, muss in erster Linie die Akkulaufzeit optimiert werden, da diese bei dem hier vorgestellten Konzept das größte Ressourcenproblem darstellt. Zwar wurden bereits im Rahmen der hier vorliegenden Arbeit Optimierungen umgesetzt. Beispielsweise wird die Häufigkeit, mit der das Smartphone seine Geoposition bezieht, und welche danach vom Algorithmus verarbeitet wird, reduziert sobald der Zustand „Auto“ festgestellt wurde. Dadurch wird die Akkulaufzeit verlängert. Anders sieht es jedoch aktuell bei den Zuständen „zu Fuß“, „U-Bahn“, „S-Bahn“, „Tram“, „Zug“ und „Bus“ aus. Der hier vorgestellte Prototyp erhebt alle eine bis fünf Sekunden die Geoposition und verarbeitet diese unabhängig davon, ob eine so häufige Abtastrate sinnvoll ist oder nicht. Es muss untersucht werden, welche Abtastrate für ein intNAV sinnvoll ist. Um also eine Optimierung hinsichtlich der Ressourcenverwaltung zu gewährleisten, sollte anhand von Simulationen die beste Abtastfrequenz für den hier vorliegenden Algorithmus gefunden werden. Dabei sollte diskutiert werden, in wieweit die Abtastfrequenz zustandsabhängig ist.

Der hier vorgestellte Ansatz funktioniert aktuell ausschließlich für die Stadt München. Das Ziel der vorliegenden Arbeit lag darin, das Key-Point-Konzept für die Zustandserkennung von intermodalen Zuständen zu erproben. Dazu wurde eine manuell erhobene Key-Point-Datenbank für München erstellt und in dem Prototypen hinterlegt. Da die Funktionsweise basierend auf einem regelbasierten Ansatz erfolgreich nachgewiesen wurde, besteht der nächste Schritt für weiterführende Arbeiten darin, den entwickelten Key-Point-Ansatz auf weitere Städte auszuweiten. Natürlich könnte dazu ebenfalls eine manuelle Key-Point-Datenbank für jede Stadt erstellt werden. Schneller und effizienter wäre es jedoch, die Key-Points automatisiert für alle Städte mittels Big Data zu bekommen und aktuell zu halten. Menschenansammlungen könnten als Indiz dafür herangezogen werden, dass es sich dabei um Key-Points handelt. Überall dort, wo also eine Häufung von verschiedenen Mobiltelefonen mit der gleichen Geo-Position festgestellt wird, wird ein Key-Point angenommen. Diese Ansätze erlauben es auch, die Key-Point-Datenbank immer aktuell zu halten - im Gegensatz zu dem hier vorgestellten Ansatz, bei dem Änderungen manuell aktualisiert werden müssen. Die Herausforderung dabei wäre jedoch,

Menschenansammlungen, wie diese beispielsweise an einer Ampel oder in Geschäften anzutreffen sind, zu filtern, da diese keine Key-Points im Sinne des hier vorgestellten intNAV-Konzepts darstellen.

Die Informationen, welche dem Benutzer des intNAVs angezeigt werden, dienen im Rahmen der vorgestellten Arbeit ausschließlich dazu, den Benutzer darüber zu informieren, ob es auf seiner aktuell gewählten intermodalen Route zu Problemen kommt oder nicht. Zusätzlich werden Alternativrouten angezeigt, falls es zu Störungen auf der Route kommt. Ein Vorschlag für weiterführende Arbeiten wäre es, an der Verbesserung der Alternativroutenempfehlung zu arbeiten und diese an die Umweltereignisse anzupassen. Im ersten Schritt könnte das Wetter in die Alternativplanung einbezogen werden. Beispielsweise sollten intermodale Routen mit langen Gehphasen bei Regen ausgeblendet werden oder es sollte explizit in einem Dialogfenster darauf hingewiesen werden. Eine Kosten-Nutzen-Abwägung würde den Nutzer darauf hinweisen, wie viel länger dieser zwar länger unterwegs wäre, dafür aber mit hoher Wahrscheinlichkeit trocken zu seinem Termin gelangen würde. Die Geo-Daten, welche die Anwendung benötigt, können herangezogen werden, um das Wetter an diesem Ort mithilfe einer speziellen Wetteranwendung zu implementieren. Neben dem Wetter können auch weitere Umweltereignisse für die Alternativplanung von Interesse sein. Beispielsweise sollten Linien auf intermodalen Routen gemieden werden, wenn bekannt ist, dass auf diesen ein großes Menschaufkommen aufgrund von Großereignissen wie Fußballspielen, Konzerten oder Stadtfesten stattfindet. Auch hier würde das System dem Benutzer anzeigen, um wie viel langsamer dieser eventuell unterwegs wäre, wenn dieser einen Umweg fahren würde, jedoch würde der Benutzer dem Gedrängel oder der Gefahr, überhaupt erst in die Bahn hineinzukommen, davonkommen. Diese Informationen werden in einigen Städten den ÖPNV-Kunden mitgeteilt. In München wird diese Information unter anderem über den Twitter-Kanal der MVG verbreitet und kann so für die entsprechende Linie auf der intermodalen Route extrahiert werden.

LITERATURVERZEICHNIS

Alves, A, Smith, R J and Williams, L. 2013. *Getting Started with Oracle Event Processing 11G*. s.l.: Packt Publishing, 2013.

2014. Anzahl der weltweiten Carsharing-Nutzer von 2006 bis 2014 (in Millionen). *Das Statistik-Portal*. [Online] 2014. [Cited: Januar 19, 2016.] <http://de.statista.com/statistik/daten/studie/388022/umfrage/anzahl-der-weltweiten-carsharing-nutzer/>.

Beutler, Felix. 2004. Intermodalität, Multimodalität und Urbanibilty – Vision für einen nachhaltigen Stadtverkehr. *Discussion Paper SP III 2004-107*. 2004, Wissenschaftszentrum Berlin für Sozialforschung (2004).

Bolbol, A., et al. 2012. Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification. *Comput. Environ. Urban Syst.* 2012, 36, pp. 526–537.

Bosch. 2017. Bosch Sensortec. [Online] 2017. [Cited: 03 05, 2017.] https://www.bosch-sensortec.com/bst/products/all_products/bmp280.

Brown, Robert G., Meyer, Richard F. and D'Esopo, D. A. 1961. The Fundamental Theorem of Exponential Smoothing. *Operations Research*. 9, 1961, pp. 673-687.

Burns, R and Dunkel, J. 2010. *Event-Driven Architecture: Softwarearchitektur für ereignisgesteuerte Geschäftsprozesse*. Heidelberg : Xpert.press, Springer Berlin, 2010.

Byon, Y. and Liang, S. 2014. Real-time transportation mode detection using smartphones and artificial neural networks: Performance comparisons between smartphones and conventional global positioning system sensors. *J.Intell. Transp. Syst.* 2014, 18, pp. 264–272.

car2go. 2016. Free-floating Carsharing von car2go verbessert die Lebensqualität in Großstädten. [Online] 2016. https://www.car2go.com/media/data/germany/microsite-press/files/20161013_free-floating-carsharing-von-car2go-verbessert-die-lebensqualitaet.pdf.

car2go, Unternehmenskommunikation. 2016. car2go. *Fact Sheet car2go*. [Online] Juni 2016. [Cited: Januar 17, 2017.] https://www.car2go.com/media/data/germany/microsite-press/files/fact_sheet_car2go_juni_2016_de.pdf.

Chandy, M K, Etzion, O and von Ammon, R. 2011. The Event Processing Manifesto. *Event Processing*. 2011.

Chen, C., et al. 2010. .Evaluatingthefeasibilityofapassivetravelsurveycollection in a complex urban environment: Lessons learned from the New York City case study. *Transp. Res. Part A Policy Pract.* 44, 2010, pp. 830-840.

Chen, J. and Bierlaire, M. 2015. Probabilistic multimodal map-matching with rich smartphone data. *Journal of Intelligent Transportation Systems*. 2, 2015, Vol. 19, pp. 134-148.

Chlond, B. and Manz, W. 2000. *Das Mobilitätspanel für den Fernverkehr*. Karlsruhe : INVERMO, 2000.

Corporation, Oracle. 2008. *Oracle Complex Event Processing Performance, An Oracle White Paper*. Zurich : s.n., 2008.

Cugola, G and Margara, A. 2012. Processing Flows of Information From Data Stream to Complex Event Processing. *ACM Comput. Surv.* 44, 2012, pp. 15:1 - 15:62.

Eckert, M and Bry, F. 2009. *Aktuelles Schlagwort "Complex Event Processing (CEP)".* 2009.

Etzion, O and Niblett, P. 2010. *Event Processing in Action.* Greenwich, CT, USA : Manning Publications Co, 2010.

Fang, Shih-Hau, et al. 2016. Transportation Modes Classification Using Sensors. *Sensors.* 16, 2016.

Feick, Katja. 2013. Carsharing-Verbraucher-Studie: Interesse an der Nutzung eines Fremdfahrzeugs höher als das eigene Auto an andere zu vermieten. *Frost & Sullivan.* [Online] Februar 25, 2013. [Cited: Januar 08, 2016.] <http://www.frost.com/prod/servlet/press-release.pag?docid=274432545>.

Feng, Tao and Timmermans, Harry J.P. 2013. Transportation mode recognition using GPS and accelerometer. [ed.] Elsevier. *Transportation Research Part C.* 2013, pp. 118-130.

Fülöp, L J, et al. 2010. Survey on Complex Event Processing and Predictive Analytics. *Nokia Siemens Networks.* 2010.

Gong, H., et al. 2012. A GPS/GIS method for travel mode detection in New York City. *Comput. Environ. Urban Syst.* 36, 2012, pp. 131-139.

Gong, L., et al. 2014. Deriving Personal Trip Data from GPS Data: A Literature Review on the Existing Methodologies. *Procedia Soc. Behav. Sci.* 138, 2014, pp. 557-565.

Gonzalez, P.A., et al. 2009. Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks. *IET Intell. Transp. Syst.* 4, 2009, pp. 37-49.

Haane, Lavinia and Goderski, Karina . 2015. Wirtschaftswachstum führt zu mehr Staus in drei Viertel der deutschen Städte. *INRIX.* [Online] August 24, 2015. [Cited: Januar 18, 2016.] <http://inrix.com/press/scorecard-report-germany-german/>.

Hinze, A, Sachs, K and Buchmann, A. 2009. Event-Driven Applications and Enabling Technologies. *Proceedings of the Third ACM International Conferenc on Distributed Events-Based Systems, DEBS '09.* 2009, pp. 1:1 - 1:15.

Hunsicker, Frank, et al. 2012. *Trends 2030 Mobilität und Logistik.* Berlin : Innovationszentrum für Mobilität und gesellschaftlichen Wandel (InnoZ) GmbH, 2012.

Inc., Esper Team and EsperTech. 2017. Esper Reference Version 5.2.0. [Online] 2017. [Cited: 08 14, 2017.] <http://www.espertech.com/esper/release-5.2.0/esper-reference/html/index.html>.

Inc., Esper Tech. 2015. Esper: Event Processing for Java. [Online] 2015. [Cited: 04 27, 2017.] <http://www.espertech.com/products/esper.php>.

Kube, Tilman. 2013. APCOA Parking. *APCOA Parking Studie: Die Parkplatz-Suche verschlingt viel Zeit und Geld.* [Online] August 15, 2013. [Cited: Januar 22, 2016.] <http://www.apcoa.de/nachrichten/artikel/apcoa-parking-studie-die-parkplatz-suche-verschlingt-viel-zeit-und-geld.html>.

Lari, Z. and Golroo, A. 2015. Automated Transportation Mode Detection Using Smart Phone Applications via Machine Learning: Case Study Mega City of Tehran. *Proceedings of the Transportation Research Board 94th Annual Meeting.* 2015.

Loose, Willi. 2013. Car-Sharing Boom hält an. *Bundesverband Car-Sharing e.V.* [Online] 2013. [Cited: Januar 18, 2016.] <http://www.carsharing.de/presse/pressemitteilungen/carsharing-boom-haelt-an>.

- Luckham, D C. 2001.** *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Boston, MA, USA : Addison-Wesley Longman Publishing, 2001.
- Luckham, David and Schulte, Roy. 2008.** Event Processing Glossary - Version 1.1. [Online] Juli 2008. [Cited: Juni 14, 2017.] <http://complexevents.com/wp-content/uploads/2008/08/epts-glossary-v11.pdf>.
- Macias, E., Suarez, A. and Lloret, J. 2013.** Mobile sensing systems. *Sensors*. 2013, 13, pp. 17292–17321.
- Magid, Y, et al. 2010.** Industry Experience with the IBM Active Middleware Technology (AMiT) Complex Event Processing Engine. [ed.] ACM. *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems, DEBS '10*. 2010, pp. 140-149.
- Manzoni, V., et al. 2011.** Transportation mode identification and real-time co2 emission estimation using smartphones. *SENSEable City Lab*. 2011.
- Map, Open Street. 2014.** OpenStreetMap powers map data on hundreds of web sites, mobile apps, and hardware devices. [Online] 2014. [Cited: Januar 27, 2017.] <https://www.openstreetmap.org/about>.
- Markowetz, Alexander. 2015.** *Digitaler Burnout: Warum unsere permanente Smartphone-Nutzung gefährlich ist*. s.l. : Droemer eBook, 2015.
- Meyer, Claas Holger. 2013.** *Vernetzung der Verkehrsträger*. Berlin : s.n., 2013. Web 2.0 – Die neue Form multimodaler Vernetzung.
- Michelson, B. 2006.** *Event-Driven Architecture Overview*. s.l. : Patricia Seybold Group, 2006.
- Montoya, David, Abiteboul, Serge and Senellart, Pierre. 2015.** Hup-Me: Inferring and Reconciling a Timeline of User Activity from Rich Smartphone Data. *International Conference on Advances in Geographic Information Systems*. 2015.
- Murakami, E., Wagner, D. and Neumeister, D. 1997.** Using global positioning systems and personal digital assistants for personal travel surveys in the United States. *Int. Conf. Transport Survey Quality and Innovation*. 1997.
- Muralidharan, K., et al. 2014.** Barometric phone sensors: more hype than hope! *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*. 15, 2014, p. 12.
- Noeren, Dominik, et al. 2013.** *Neue Mobilität. Ein Ausblick zur urbanen Mobilität in vier Dekaden*. Freiburg : Fraunhofer ISE, 2013.
- OSM2PQSQL. 2014.** osm2pqsql is a tool for loading OpenStreetMap data into a PostgreSQL / PostGIS database. [Online] 2014. [Cited: Januar 27, 2017.] <https://github.com/openstreetmap/osm2pqsql>.
- Partsch, Ina, et al. 2012.** Positioning in Real-Time Public Transport Navigation: Comparison of Vehicle-Based and Smartphone-Generated Acceleration Data to Determine Motion States of Passengers. *Mobility: Economy - Ecology - Technology*. 6th International Symposium, 2012.
- Patterson, Donald J., et al. 2003.** Inferring High-Level Behavior from Low-Level Sensors. [book auth.] A. K. Dey, A. Schmidt and J. McCarthy. *UbiComp 2003: Ubiquitous Computing*. Berlin Heidelberg : Springer, 2003, Vol. vol. 2864 of Lecture Notes in Computer Science, pp. 73-89.
- PostgreSQL. 2014.** PostgreSQL, the world's most advanced open source database. [Online] 2014. [Cited: Januar 27, 2017.] <http://www.postgresql.org>.

Reddy, Sasank, et al. 2008. Determining Transportation Mode On Mobile Phones. *Proceedings of the 2008 12th IEEE International Symposium on Wearable Computers*. 2008, ISWC '08, pp. 25-28.

Reddy, Sasank, et al. 2010. Using Mobile Phones to Determine. *ACM Trans. Sensor Network*. 6, 2010, Vol. 13.

—. **2010.** Using Mobile Phones to Determine Transportation Modes. *ACM Trans Sen. Netw.* . March 2010, Vol. 6, 2, pp. 13:1-13:27.

Reichel, Bernd. 2015. 2014: Fahrgastaufkommen in Bussen und Bahnen weiter auf Wachstumskurs. *DeStatis*. [Online] April 07, 2015. [Cited: Januar 18, 2016.] https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2015/04/PD15_121_461.html.

Schalk, Thiemo. 2014. Pressemeldung. *BMW Group PressClub Deutschland*. [Online] Juli 03, 2014. [Cited: Januar 12, 2016.] https://www.press.bmwgroup.com/deutschland/pressDetail.html?title=car-sharing-als-wegbereiter-f%C3%BCr-elektromobilit%C3%A4t-zwischenergebnisse-des-forschungsprojekts-wimobil&outputChannelId=7&id=T0185946DE&left_menu_item=node__5248.

Schuessler, N. and Axhausen, K. 2009. Processing Raw Data from Global Positioning Systems without Additional Information. *Transp. Res. Rec. J. Transp. Res. Board*. 2105, 2009, pp. 28-36.

Shen, L. and Stopher, P. 2014. Review of GPS travel survey and GPS data-processing methods. *Transp. Rev.* 34, 2014, pp. 316-334.

Siirtola, Pekka and Rönig, Juha. 2012. Recognizing Human Activities User-independently on Smartphones Based on Accelerometer Data. *International Journal of Interactive Multimedia and Artificial Intelligence*. 5, 2012, pp. 38-45.

Smith, Aaron. 2013. Smartphone Ownership – 2013 Update. *PewResearchCenter*. [Online] Juni 5, 2013. [Cited: Januar 29, 2017.] http://boletines.prisadigital.com/PIP_Smartphone_adoption_2013.pdf.

Sonderen, T. 2016. *Detection of transportation mode solely using smartphones*. University of Twente : s.n., 2016.

statista. 2012. Pkw-Dichte in ausgewählten weltweiten Ländern im Jahr 2012 (Pkw je 1.000 Einwohner). *Das Statistik-Portal*. [Online] 2012. [Cited: Januar 31, 2016.] <http://de.statista.com/statistik/daten/studie/163407/umfrage/pkw-dichte-in-ausgewaehlten-laendern/>.

Stenneth, Leon, et al. 2011. Transportation Mode Detection using Mobile Phones and GIS Information. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 2011, pp. 54-63.

Stipkovic, S. 2012. Complex Event Processing auf mobilen Systemen. *Masterarbeit*. 2012.

Stopher, P., et al. Deducing Mode and Purpose from GPS Data. <http://ws.econ.usyd.edu.au/itls/wp-archive/itls-wp-08-06.pdf>. [Online] [Cited: November 9, 2016.]

Su, Xing, et al. 2014. Travel Mode Identification with Smartphones. *Transportation Research Board*. 94, 2014, pp. 15-4871.

Swiaczny, Frank. 2008. Urbanisierung und Entwicklung 2008. *Zeitschrift für Bevölkerungswissenschaft*. Volume 33, Oktober 2008, Issue 3, pp. 449-459.

Systems, StreamBase. 2010. StreamBase Named 2010 World economic Forum Technology Pioneer. [Online] 2010. [Cited: 08 14, 2017.] <http://www.streambase.com/news-and-events/press-releases/pr-2009/streambase-named2010-world-economic-forum-technology-pioneer>.

- Tibbetts, R. 2009.** Performance & Scalability Characterization. *Stream Based Systems*. [Online] 2009. [Cited: 08 2017, 14.] <http://www.streambase.com/news-and-events/press-releases/pr-2009/streambase-named-2010-world-economic-forum-technology-pioneer/>.
- Vasseur, A. 2014.** Esper Performance. [Online] 2014. [Cited: April 9, 2015.] <http://docs.codehaus.org/display/ESPER/Esper+performance>.
- Vidackovic, K, Renner, T and Rex, S. 2010.** Marktübersicht RealTime Monitoring Software, Event Processing Tools im Überblick. [Online] 2010. [Cited: April 09, 2015.] <http://wiki.iao.frauenhofer.de/images/studien/marktuebersicht-real-time-monitoring-software-frauenhofer-iao.pdf>.
- von der Ruhren, S., et al. 2003.** *Bestimmung multimodaler Personengruppen*. Aachen : s.n., 2003.
- von Nauman, Aurika. 2016.** Pressemitteilung. *Jahresbilanz 2016: DriveNow weiter auf Erfolgskurs*. [Online] Januar 14, 2016. [Cited: Januar 30, 2017.] https://content.drive-now.com/sites/default/files/2017-03/2017.01.19_DriveNow_Jahresbilanz_2016.pdf.
- Widhalm, Peter, Nitsche, Philippe and Brändle, Norbert. 2012.** Transport Mode Detection with Realistic Smartphone Sensor Data. *21st International Conference on Pattern Recognition*. 2012, pp. 573-576.
- Wu, Linlin, Yang, Biao and Jing, Peng. 2016.** Travel Mode Detection Based on GPS Raw Data Collected by Smartphones: A Systematic Review of the Existing Methodologies. *Information*. 7, 2016, 67.
- Xia, H., et al. 2014.** Using Smart Phone Sensors to Detect Transportation Modes. *sensors*. 2014, pp. 20843-20865.
- Xiao, G., Juan, Z. and Gao, J. 2015.** Inferring Trip Ends from GPS Data Based on Smartphones in Shanghai. *Proceedings of the Transportation Research Board 94th Annual Meeting*. 2015.
- . 2015. Travel Mode Detection Based on Neural Networks and Particle Swarm Optimization. *Information*. 6, 2015, pp. 522-535.
- Xiao, G., Juan, Z. and Zhang, C. 2015.** Travel mode detection based on GPS track data and Bayesian networks. *Comput. Environ. Urban Syst*. 54, 2015, pp. 14-22.
- Xu, Chao, et al. 2010.** Identifying Travel Mode from GPS Trajectories through Fuzzy Pattern Recognition. *Seventh International Conference on Fuzzy Systems and Knowledge Discovery*. 2010, pp. 889-893.
- Yang, F., Yao, Z. and Jin, P. 2015.** GPS and Acceleration Data in Multimode Trip Data Recognition Based on Wavelet Transform Modulus Maximum Algorithm. *Transp. Res. Rec. J. Transp. Res. Board*. 2526, 2015, pp. 90-98.
- Yue, Y., et al. 2014.** Zooming into individuals to understand the collective: A review of trajectory-based travel behavior studies. *Travel Behav. Soc*. 1, 2014, pp. 69-78.
- Zhang, L., et al. 2011.** Multi-Stage Approach to Travel-Mode Segmentation and Classification of GPS Traces. *Proceedings of the ISPRS Guilin 2011 Workshop on International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2011.
- Zheng, Y., et al. 2008.** Understanding mobility based on GPS data. *Ubiquitous Computing*. 2008, pp. 312-321.

ABBILDUNGSVERZEICHNIS

Abbildung 1: Einflussfaktoren auf die Mobilität der Zukunft (Meyer, 2013)	4
Abbildung 2: Informationsdefizite innerhalb intermodaler Routen	6
Abbildung 3: Differenzierung der Begriffe Multimodalität und Intermodalität (von der Ruhren, et al., 2003) ...	14
Abbildung 4: Mono- und Intermodalität als Sonderformen der Multimodalität	15
Abbildung 5: Navigation als Summe der Bereiche "Routing" und "Guiding"	16
Abbildung 6: Vereinfachte Übersicht der Funktionalität eines intNAVs. Die Zahlen 1-6 entsprechen dabei den Anforderungen 1-6.....	19
Abbildung 7: Bestandteile eines intNAVs und Einordnung des Forschungsschwerpunktes der Doktorarbeit.....	21
Abbildung 8: GPS-Logs der Fortbewegungsarten Bus (A), zu Fuß (B) und Auto (C) (Gonzalez, et al., 2009).....	24
Abbildung 9: <x,y,z>-Achsen eines Smartphones	25
Abbildung 10: Beschleunigungserkennung für die Zustände Gehen (rot) und Bus (blau) (Su, et al., 2014)	26
Abbildung 11: Unterschiede in der Luftdruckerkennung bei den Transportmodi Auto (gelb), Bus (rot), U-Bahn (blau) (Su, et al., 2014)	27
Abbildung 12: Aufzeichnungen der Transportmodi mittels magnetischem Feld (Su, et al., 2014)	28
Abbildung 13: Verteilung der Roh-Daten auf der X-Achse von drei Sensoren zur Bestimmung des Transportmittels (Fang, et al., 2016).....	31
Abbildung 14: Erkennungsraten von unterschiedlichen Transportmitteln nach Stenneth et al. (Stenneth, et al., 2011)	33
Abbildung 15: Erkennungsrate der Transportmittel nach Montaya (Montoya, et al., 2015).....	33
Abbildung 16: Abstrakte KP-Graphen-Visualisierung und Darstellung des Bezugs zwischen mehreren KPs, um die öffentliche Infrastruktur in München abzubilden	38
Abbildung 17: Zustandsautomat der oberen Logik	42
Abbildung 18: Logik: Unbekannt --> Gehen/ Motorisiert	42
Abbildung 19: Abbruchbedingung	43
Abbildung 20: Logik: Motorisiert --> Auto	44
Abbildung 21: Logik: Gehen --> Motorisiert --> ÖPNV.....	44
Abbildung 22: KP-Verhalten der unteren Logik	45
Abbildung 23: Logik ÖPNV <--> Transition.....	46
Abbildung 24: Mögliche Transitionen (AP = Aktueller Punkt; ZP = Zielpunkt).....	46
Abbildung 25: Zustandsautomat der unteren Logik	48
Abbildung 26: Zusammenhang zwischen Oberer und Unterer Logik	49
Abbildung 27: Beispiel einer Graphensuche mit niedriger Abtastrate	51
Abbildung 28: Abstraktes Datenmodell zur Repräsentation von KPs, Stationen und Linien.....	58
Abbildung 29: Routenverlauf des Testlaufs	60
Abbildung 30: Darstellung möglicher Situationen während der Verwendung einer intermodalen Route	66
Abbildung 31: Auszug einer intermodalen Route.....	69
Abbildung 32: Arten von Deviationen	74
Abbildung 33: Überblick über die Architektur zur Prognose von Anomalien.....	76
Abbildung 34: Wahrscheinlichkeit einer Abweichung in der Raumdimension	79
Abbildung 35: Wahrscheinlichkeit einer Abweichung in der Zeitdimension.....	82
Abbildung 36: Schlussfolgerung basierend auf einem Multi-Schichten Neuronalen Netzwerk	87
Abbildung 37: Lernen mit Hilfe der Rückwärts-Propagation	89
Abbildung 38: Events verschiedener Abstraktionslevel. Events auf der unteren Abstraktionsebene sind einfache Events. Komplexe Events können im höheren Abstraktionslevel gefunden werden. (nach (Burns, et al., 2010))	94
Abbildung 39: Logische Schichten eines typischen EDA: Die Eventverarbeitungsschicht (Event Processing Layer) beinhaltet einen oder mehrere CEPs und erhält Events aus unterschiedlichen Quellen. Die resultierenden komplexen Events werden weitergeleitet an die Event Handling Schicht um dort weitere Aktionen auszuführen (nach (Burns, et al., 2010)).	95

Abbildung 40: Elemente von EPA: Events werden durch die EPA mittels der Input und Output Ströme geschickt. Die Events werden mit Hilfe der sog. Eventmodelle (Event Models) definiert. Der Eventstrom wird mit den Eventregeln innerhalb der Eventverarbeitungseinheit (Event Processing Engine) verglichen. Die Events können um weitere Informationen aus der externen Datenbank angereichert werden (nach (Burns, et al., 2010))	97
Abbildung 41: (N)Esper Architektur (Inc., 2015)	99
Abbildung 42: Beispiel für ein einfaches Statement <code>select * from Withdrawal</code> . Die linke Zeitleiste zeigt eine Serie von Abhebe-Events 1 bis 6 an, welche über die Zeit anfallen. Die Höhe die abgehoben wird, steht dabei jeweils in runden Klammern. Die zwei Spalten auf der rechten Seite zeigen, dass nur neue Events und keine alten Events berücksichtigt werden ((Inc., 2017), Seite 38).	102
Abbildung 43: Beispiel für das Statement <code>select * from Withdrawal.win:length(N)</code> . Alle ankommenden Events werden als neue Events auf dem Listener berücksichtigt. Wenn das Event W1 das Fenster aufgrund der Ankunft von W6 verlässt, wird es als altes Event auf dem Listener eingefügt ((Inc., 2017), Seite 39).	103
Abbildung 44: Beispiel des Statements <code>select * from Withdrawal.win:time(4sec)</code> . Das Diagramm startet zu einer bestimmten Zeit t und zeigt die Inhalte um t+4, t+5 usw. Das Event W1 wird beim Listener um t+4 als neues Event gemeldet. Wenn t+8 eintritt, verlässt dieses Event das Fenster t und wird als als erstes old Event zum Update Listener gemeldet ((Inc., 2017), Seite 43)	104
Abbildung 45: CEP-Architektur eines intNAVs	108
Abbildung 46: Eine Route aus Google Directions API aufgeteilt in „Legs“ und „Steps“.	111
Abbildung 47: CEP auf einem mobilen System und auf einem Server. Wenn die CEP auf einen Server ausgelagert wird, ist die Verbindung nicht zuverlässig und Daten können verloren gehen (nach (Stipkovic, 2012)).	112
Abbildung 48: UML Diagramm zeigt die Events, welche in dem Prototypen verwendet wurden.	115
Abbildung 49: Screenshot des Java Prototypen welcher auf einem lokalen Computer und nicht auf einem mobilen Endgerät entwickelt wurde. Dabei wurden nur simulierte Daten verwendet. Der Prototyp wurde entwickelt um zu testen ob komplexe Events in intNAVs entdeckt werden können und die entsprechenden Empfehlungen ausgegeben werden.	117
Abbildung 50: Wenn die Applikation zum ersten Mal ausgeführt wird, erscheint folgender Informationsdialog.	118
Abbildung 51: SplashScreenActivity: diese Aktivität wird aufgerufen, wenn die Applikation gestartet wird. Wenn GPS bei Applikationsstart aktiviert ist, wird die aktuelle Position in der Datenbank abgespeichert.	119
Abbildung 52: PlanRouteActivity: diese Aktivität wird benötigt um eine Route mittels Google Directions API zu planen.	119
Abbildung 53: RouteGuideActivity: diese Aktivität wird während der gesamten intermodalen Navigation gezeigt. Auf der linken Seite sieht man die geplante Route und der markierte grüne Abschnitt zeigt an, in welchem Abschnitt der intermodalen Route sich der Benutzer befindet.	120
Abbildung 54: RouteGuideActivity: Ein Warndreieck im ersten Abschnitt informiert den Benutzer über ein Problem auf diesem Abschnitt. Wenn das Warndreieck angeklickt wird, wird die entsprechende Information angezeigt.	121
Abbildung 55: Für jede Empfehlung wird eine Android Benachrichtigung generiert.	121
Abbildung 56: FinalFeedbackActivity: diese Aktivität wird ausgeführt, um von dem Benutzer ein Feedback über die GUI des Android Prototypen zu erhalten.	122
Abbildung 57: SettingsActivity: der Benutzer kann seine Einstellung in der Applikation ändern.	123
Abbildung 58: Meistverwendete Verkehrsmittel Innerorts in München (1 - nie bis 6 - jeden Tag)	131
Abbildung 59: Erfahrung mit dem Umgang von mobilen Endgeräten	132
Abbildung 60: Erfahrung mit intermodalen Navigationssystemen	132
Abbildung 61: Gegenüberstellung der Umfrageergebnisse bzgl. der Anzeige des Inhalts von Störungen (0 – unwichtig bis 6 – sehr wichtig)	137
Abbildung 62: Gegenüberstellung der Präferenz bzgl. der Benachrichtigungsintervalle von aktiven und passiven Störungen	139
Abbildung 63: Situationsabhängig Benachrichtigung 1 – nicht störend ... 6 – sehr störend)	140
Abbildung 64: Verteilung der Ergebnisse zur Transportmittelerkennung in Echtzeit	145

Abbildung 65: Verteilung der Ergebnisse zur Anomalieerkennung.....	145
Abbildung 66: Verteilung der Ergebnisse zur Performanz der Applikation	146
Abbildung 67: Verteilung der Ergebnisse zur Kommunikation von Störungen	146
Abbildung 68: Verteilung der Ergebnisse zu den gezeigten Empfehlungen	147
Abbildung 69: Verteilung der Ergebnisse zur Informationsdichte.....	147
Abbildung 70: Verteilung der Ergebnisse zur Wiederkehrwahrscheinlichkeit	148
Abbildung 71: Überblick über die Situationen, in welchen das intNAV von den Probanden verwendet wurde	149
Abbildung 72: Überblick über Situationen in denen die Applikation hilfreich war	149
Abbildung 73: Überblick über Situationen in denen die Applikation überflüssig erschien	150
Abbildung 74: Punkte, welche den Probanden positiv im Gedächtnis geblieben sind	151
Abbildung 75: Punkte, welche den Probanden negativ im Gedächtnis geblieben sind	151
Abbildung 76: Vergleich der Funktionen eines intNAVs mit einem traditionellen Navigationssystem.....	152
Abbildung 77: Verteilung der Erwartungserfüllung.....	153

TABELLENVERZEICHNIS

Tabelle 1: Gegenüberstellung der Erkennungswahrscheinlichkeiten für verschiedene Transportmodi	29
Tabelle 2: Überblick über mögliche KP-Typen und die Rückschlüsse bezüglich der TME	38
Tabelle 3: Mögliche Transitionen zwischen den Modi	40
Tabelle 4: SQLite-Tabelle zum Speichern von Stationen-KPs	55
Tabelle 5: SQLite-Tabelle für Linien	56
Tabelle 6: SQLite-Tabelle zum Speichern des Graphen zwischen den KPs.	56
Tabelle 7: SQLite-Tabelle beinhaltet die Kanten und die entsprechenden Linien aus dem ÖPNV	56
Tabelle 8: SQLite Tabelle speichert verschiedene Typen um die KPs detaillierter beschreiben zu können.	56
Tabelle 9: Mögliche Situationen vor einem Routenabschnitt	71
Tabelle 10: Mögliche Situationen während eines Routenabschnittes	71
Tabelle 11: Überblick über die verwendeten Events im Prototypen	114
Tabelle 12: Vergleich möglicher Eventrepräsentationen in Esper ((Inc., 2017), Seite 36)	116
Tabelle 13: Esper Event Server und Client Konfiguration	125
Tabelle 14: Einstellung bzgl. GPS	132
Tabelle 15: Kriterien eines intNAVs (Auszug aus Appendix A)	134
Tabelle 16: Ergebnisse der Befragung zu Informationen bei erkannten aktiven Störungen (Auszug aus Appendix A)	136
Tabelle 17: Ergebnisse der Befragung zu Informationen bei erkannten passiven Störungen (Auszug aus Appendix A)	137
Tabelle 18: Umfrageergebnisse bzgl. Benachrichtigungsdauer bei aktiven (links) und passiven (rechts) Störungen (Auszug aus Appendix A)	138
Tabelle 19: Umfrageergebnisse bzgl. intervallartiger Benachrichtigungsart bei aktiven (links) und passiven (rechts) Störungen (Auszug aus Appendix A)	138
Tabelle 20: Bewertung der sieben Messgrößen mit einer positiven und negativen Tendenz	144

227818 Umfrage Intermodaler Routenplaner

Die Teilnehmer können nicht teilnehmen. ⓘ

02.08.2015 13:36:17

► Legende

Frage: Geschlecht

	ANZAHL	PROZENT	
männlich (1)	21	61.76%	
weiblich (2)	13	38.24%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	1.38		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Was sind die Verkehrsmittel, die Sie innerorts am häufigsten benutzen?

Bitte ordnen Sie die Verkehrsmittel nach der Zeit ein, die Sie sie benutzen, um sich innerorts fortzubewegen.

	0 - NIE (1)	1 (2)	2 (3)	3 (4)	4 (5)	5 - JEDEN TAG (6)	GESAMT	MITTELWERT	MISSING*	
									A	B
Auto	5.88% (2)	17.65% (6)	17.65% (6)	23.53% (8)	17.65% (6)	17.65% (6)	34	3.82	0	0
Motorrad/ Roller	85.29% (29)	2.94% (1)	5.88% (2)	5.88% (2)	0.00% (0)	0.00% (0)	34	1.32	0	0
öffentliche Verkehrsmittel	0.00% (0)	18.18% (6)	21.21% (7)	12.12% (4)	24.24% (8)	24.24% (8)	33	4.15	0	1
Fahrrad	14.71% (5)	17.65% (6)	35.29% (12)	17.65% (6)	5.88% (2)	8.82% (3)	34	3.09	0	0
zu Fuß	0.00% (0)	6.06% (2)	24.24% (8)	24.24% (8)	15.15% (5)	30.30% (10)	33	4.39	0	1
Sonstiges (z.B. OneWheel)	87.10% (27)	6.45% (2)	3.23% (1)	3.23% (1)	0.00% (0)	0.00% (0)	31	1.23	0	3

N = 34 | n = 34 | SYS-MISSING = 0

* A = N/A B = UNGÜLTIG (FEHLEND)

Frage: Haben Sie Erfahrung im Umgang mit Smartphones?

	ANZAHL	PROZENT	
Ja (1)	32	94.12%	
Nein (2)	2	5.88%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	1.06		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Haben Sie Erfahrung in der Benutzung von Standortdiensten, Routenplanern oder Navigationssystemen?
z.B. Navigationssysteme im Auto oder Google Maps

	ANZAHL	PROZENT	
Ja (1)	34	100.00%	
Nein (2)	0	0.00%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	1.00		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Haben Sie Erfahrung im Umgang mit intermodalen Navigationssystemen?

	ANZAHL	PROZENT	
Ja (1)	18	52.94%	
Nein (2)	16	47.06%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	1.47		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Haben Sie Erfahrung im Umgang mit intermodalen Navigationssystemen auf Smartphones?

	ANZAHL	PROZENT	
Ja (1)	16	47.06%	
Nein (2)	18	52.94%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	1.53		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Die Applikation informiert sie bereits ein paar Minuten vorher, dass Sie bald Ihre Route antreten müssen. Wie weit im Voraus möchten Sie eine Benachrichtigung erhalten?

	ANZAHL	PROZENT	
1 Minute (1)	1	2.94%	
2 Minuten (2)	0	0.00%	
5 Minuten (3)	7	20.59%	
10 Minuten (4)	11	32.35%	
15 Minuten (5)	10	29.41%	
30 Minuten (6)	4	11.76%	
60 Minuten (7)	1	2.94%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	4.32		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Möchten Sie in bestimmten Intervallen Benachrichtigungen erhalten, dass Sie bald starten müssen?

	ANZAHL	PROZENT	
Ja (1)	18	52.94%	
Nein (2)	16	47.06%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	1.47		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Wenn ja, in welchen Intervallen sollten die Benachrichtigungen eintreffen?

	ANZAHL	PROZENT	
jede Minute (1)	1	5.26%	
alle 2 Minuten (2)	2	10.53%	
alle 5 Minuten (3)	6	31.58%	
alle 10 Minuten (4)	4	21.05%	
lediglich 1 Minute vor Beginn (5)	6	31.58%	
GESAMT	19		
ungültig (fehlend)	15		
Mittelwert	3.63		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Welches Signal würden Sie bevorzugen, um auf eine Benachrichtigung hingewiesen zu werden?

	ANZAHL	PROZENT	
akustisches Signal (1)	8	23.53%	
Vibrationsalarm (2)	13	38.24%	
beides (3)	13	38.24%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	2.15		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Was für eine Benachrichtigungsdauer würden Sie bevorzugen?

ANZAHL PROZENT

1 Sekunde (1)	19	55.88%	
so lange, bis eine Aktion ausgeführt wird, die die Meldung deaktiviert (Wecker-artig) (2)	15	44.12%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	1.44		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Welches Signal würden Sie bevorzugen, um auf eine Benachrichtigung hingewiesen zu werden?

	ANZAHL	PROZENT	
akustisches Signal (1)	10	29.41%	
Vibrationsalarm (2)	9	26.47%	
beides (3)	15	44.12%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	2.15		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Was für eine Benachrichtigungsdauer würden Sie bevorzugen?

	ANZAHL	PROZENT	
1 Sekunde (1)	13	38.24%	
so lange, bis eine Aktion ausgeführt wird, die die Meldung deaktiviert (Wecker-artig) (2)	21	61.76%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	1.62		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Möchten Sie so lange Benachrichtigungen erhalten bis die Störung nicht mehr besteht?

	ANZAHL	PROZENT	
Ja (1)	11	32.35%	
Nein (2)	23	67.65%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	1.68		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Wenn ja, in welchen Intervallen sollten die Benachrichtigungen eintreffen?

	ANZAHL	PROZENT	
alle 10 Sekunden (1)	1	6.67%	
alle 30 Sekunden (2)	3	20.00%	
jede Minute (3)	2	13.33%	
alle 2 Minuten (4)	2	13.33%	
alle 5 Minuten (5)	7	46.67%	
GESAMT	15		
ungültig (fehlend)	19		
Mittelwert	3.73		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Neben dem Grund für die Benachrichtigung (aktive Störung liegt vor) können weitere Informationen angezeigt werden. Nennen Sie solche, die Sie als Nutzer gerne auf einen Blick sehen würden.

	0 - UNWICHTIG (1)	1 (2)	2 (3)	3 (4)	4 (5)	5 - WICHTIG (6)	GESAMT	MITTELWERT	MIS A
Klassifizierung - um was für eine Störung es sich handelt (z.B. befindet sich im falschen öffentlichen Verkehrsmittel)	0.00% (0)	0.00% (0)	2.94% (1)	5.88% (2)	26.47% (9)	64.71% (22)	34	5.53	0
Alternativen/ Aktionsmöglichkeiten – Benutzer wird über alternative Routen informiert (z.B. anderen Bus nehmen)	0.00% (0)	2.94% (1)	8.82% (3)	20.59% (7)	23.53% (8)	44.12% (15)	34	4.97	0
Konsequenzen – Auswirkungen, wenn die Route nicht eingehalten wird (z.B. U-Bahn verpasst, nächste in 30 Minuten)	0.00% (0)	11.76% (4)	11.76% (4)	23.53% (8)	32.35% (11)	20.59% (7)	34	4.38	0

N = 34 | n = 34 | SYS-MISSING = 0

* A = N/A B = UNGÜLTIG (FEHL)

Frage: Wie störend würden Sie eine EINMALIGE Benachrichtigung zu einer aktiven Störung in folgenden Situationen betrachten?

z.B. zu spätes Losgehen

	0 - NICHT STÖREND (1) 1 (2) 2 (3) 3 (4) 4 (5) 5 - SEHR STÖREND (6)						GESAMT	MITTELWERT	MISSING*	
	A	B								
allein zu Hause	70.59% (24)	20.59% (7)	2.94% (1)	5.88% (2)	0.00% (0)	0.00% (0)	34	1.44	0	0
Projektbesprechung	8.82% (3)	14.71% (5)	14.71% (5)	8.82% (3)	23.53% (8)	29.41% (10)	34	4.12	0	0
romantisches Date	5.88% (2)	5.88% (2)	8.82% (3)	11.76% (4)	32.35% (11)	35.29% (12)	34	4.65	0	0
in einer Bar unter Freunden	23.53% (8)	26.47% (9)	26.47% (9)	17.65% (6)	5.88% (2)	0.00% (0)	34	2.56	0	0
sitzend in einem Bus	58.82% (20)	29.41% (10)	5.88% (2)	5.88% (2)	0.00% (0)	0.00% (0)	34	1.59	0	0
stehend in einer überfüllten U-Bahn	39.39% (13)	27.27% (9)	21.21% (7)	9.09% (3)	0.00% (0)	3.03% (1)	33	2.12	0	1
beim Einkaufen vor einem Regal	35.29% (12)	38.24% (13)	11.76% (4)	8.82% (3)	2.94% (1)	2.94% (1)	34	2.15	0	0
an der Kasse beim Bezahlen	17.65% (6)	14.71% (5)	23.53% (8)	14.71% (5)	11.76% (4)	17.65% (6)	34	3.41	0	0
kurz nach dem Aufwachen	26.47% (9)	26.47% (9)	17.65% (6)	11.76% (4)	2.94% (1)	14.71% (5)	34	2.82	0	0
beim Sport (z.B. Joggen)	20.59% (7)	20.59% (7)	23.53% (8)	14.71% (5)	8.82% (3)	11.76% (4)	34	3.06	0	0
beim Fernsehen	52.94% (18)	32.35% (11)	8.82% (3)	2.94% (1)	2.94% (1)	0.00% (0)	34	1.71	0	0

N = 34 | n = 34 | SYS-MISSING = 0

* A = N/A B = UNGÜLTIG (FEHLEND)

Frage: Wie störend würden Sie eine MEHRMALIGE Benachrichtigung zu einer aktiven Störung in folgenden Situationen betrachten?

Bei mehrmaligen Benachrichtigungen werden Sie in bestimmten Intervallen benachrichtigt

	0 - NICHT STÖREND (1) 1 (2) 2 (3) 3 (4) 4 (5) 5 - SEHR STÖREND (6)						GESAMT	MITTELWERT	MISSING*	
	A	B								
allein zu Hause	32.35% (11)	26.47% (9)	26.47% (9)	5.88% (2)	5.88% (2)	2.94% (1)	34	2.35	0	0
Projektbesprechung	2.94% (1)	2.94% (1)	2.94% (1)	17.65% (6)	11.76% (4)	61.76% (21)	34	5.18	0	0
romantisches Date	2.94% (1)	2.94% (1)	2.94% (1)	2.94% (1)	14.71% (5)	73.53% (25)	34	5.44	0	0
in einer Bar unter Freunden	2.94% (1)	5.88% (2)	23.53% (8)	32.35% (11)	23.53% (8)	11.76% (4)	34	4.03	0	0
sitzend in einem Bus	17.65% (6)	14.71% (5)	26.47% (9)	29.41% (10)	8.82% (3)	2.94% (1)	34	3.06	0	0
stehend in einer überfüllten U-Bahn	5.88% (2)	20.59% (7)	17.65% (6)	17.65% (6)	20.59% (7)	17.65% (6)	34	3.79	0	0
beim Einkaufen vor einem Regal	14.71% (5)	8.82% (3)	35.29% (12)	14.71% (5)	14.71% (5)	11.76% (4)	34	3.41	0	0
an der Kasse beim Bezahlen	6.06% (2)	6.06% (2)	21.21% (7)	15.15% (5)	15.15% (5)	36.36% (12)	33	4.36	0	1
kurz nach dem Aufwachen	8.82% (3)	11.76% (4)	29.41% (10)	8.82% (3)	17.65% (6)	23.53% (8)	34	3.85	0	0
beim Sport (z.B. Joggen)	8.82% (3)	2.94% (1)	23.53% (8)	29.41% (10)	17.65% (6)	17.65% (6)	34	3.97	0	0
beim Fernsehen	30.30% (10)	24.24% (8)	15.15% (5)	15.15% (5)	9.09% (3)	6.06% (2)	33	2.67	0	1

N = 34 | n = 34 | SYS-MISSING = 0

* A = N/A B = UNGÜLTIG (FEHLEND)

Frage: Wie würden Sie in der Mehrzahl der oben genannten Situationen mit den Benachrichtigungen umgehen?

	ANZAHL	PROZENT
die Benachrichtigungen nach der ersten Nachricht abschalten (1)	33	97.06%
kontinuierlich Nachrichten bekommen (2)	1	2.94%
GESAMT	34	
ungültig (fehlend)	0	
Mittelwert	1.03	

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Welches Signal würden Sie bevorzugen, um auf eine Benachrichtigung hingewiesen zu werden?

	ANZAHL	PROZENT	
akustisches Signal (1)	6	17.65%	
Vibrationsalarm (2)	11	32.35%	
beides (3)	17	50.00%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	2.32		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Was für eine Benachrichtigungsdauer würden Sie bevorzugen?

	ANZAHL	PROZENT	
1 Sekunde (1)	16	47.06%	
so lange, bis eine Aktion ausgeführt wird, die die Meldung deaktiviert (Wecker-artig) (2)	18	52.94%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	1.53		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Möchten Sie so lange Benachrichtigungen erhalten bis die Störung nicht mehr besteht?

	ANZAHL	PROZENT	
Ja (1)	10	29.41%	
Nein (2)	24	70.59%	
GESAMT	34		
ungültig (fehlend)	0		
Mittelwert	1.71		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Wenn ja, in welchen Intervallen sollten die Benachrichtigungen eintreffen?

	ANZAHL	PROZENT	
alle 10 Sekunden (1)	1	6.67%	
alle 30 Sekunden (2)	1	6.67%	
jede Minute (3)	1	6.67%	
alle 2 Minuten (4)	4	26.67%	
alle 5 Minuten (5)	8	53.33%	
GESAMT	15		
ungültig (fehlend)	19		
Mittelwert	4.13		

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Neben dem Grund für die Benachrichtigung (passive Störung liegt vor) können weitere Informationen angezeigt werden. Nennen Sie solche, die Sie als Nutzer gerne auf einen Blick sehen würden.

	0 - UNWICHTIG (1)	1 (2)	2 (3)	3 (4)	4 (5)	5 - WICHTIG (6)	GESAMT	MITTELWERT	MIS A
Klassifizierung - um was für eine Störung es sich handelt (z.B. der Bus hat eine Verspätung)	0.00% (0)	2.94% (1)	0.00% (0)	14.71% (5)	20.59% (7)	61.76% (21)	34	5.38	0
Alternativen/ Aktionsmöglichkeiten – Benutzer wird über alternative Routen informiert (z.B. anderen Bus nehmen)	0.00% (0)	0.00% (0)	8.82% (3)	17.65% (6)	23.53% (8)	50.00% (17)	34	5.15	0
Konsequenzen – Auswirkungen, wenn die Route nicht eingehalten wird (z.B. U-Bahn verpasst, nächste in 30 Minuten)	0.00% (0)	5.88% (2)	14.71% (5)	14.71% (5)	29.41% (10)	35.29% (12)	34	4.74	0

N = 34 | n = 34 | SYS-MISSING = 0

* A = N/A B = UNGÜLTIG (FEHL)

Frage: Wie bewerten Sie folgende Punkte?

Bitte geben Sie an, was für einen Detailgrad an Informationen Sie sich bei der Anzeige von Verspätungen wünschen.

	-5 - SEHR NEGATIV (1)	-4 (2)	-3 (3)	-2 (4)	-1 (5)	0 - NEUTRAL (6)	1 (7)	2 (8)	3 (9)	4 (10)	5 - SEHR POSITIV (11)	GE
Mitteilung: "Die U6 verspätet sich mit einer Wahrscheinlichkeit von 66% um 3 bis 5 Minuten."	2.94% (1)	5.88% (2)	14.71% (5)	14.71% (5)	20.59% (7)	23.53% (8)	5.88% (2)	2.94% (1)	5.88% (2)	0.00% (0)	2.94% (1)	
Mitteilung: "Die U6 verspätet sich um etwa 3 bis 5 Minuten."	0.00% (0)	0.00% (0)	0.00% (0)	2.94% (1)	0.00% (0)	5.88% (2)	5.88% (2)	14.71% (5)	20.59% (7)	20.59% (7)	29.41% (10)	

Mitteilung: "Die U6 verspätet sich um etwa 4 Minuten." 2.94% (1) 0.00% (0) 0.00% (0) 0.00% (0) 8.82% (3) 5.88% (2) 5.88% (2) 26.47% (9) 14.71% (5) 17.65% (6) 17.65% (6)

N = 34 | n = 34 | SYS-MISSING = 0

* A =

Frage: Wie störend würden Sie eine **EINMALIGE** Benachrichtigung zu einer passiven Störung in folgenden Situationen betrachten?
z.B. zu spätes Losgehen

	0 - NICHT STÖREND (1)	1 (2)	2 (3)	3 (4)	4 (5)	5 - SEHR STÖREND (6)	GESAMT	MITTELWERT	MISSING*	
									A	B
allein zu Hause	64.71% (22)	32.35% (11)	2.94% (1)	0.00% (0)	0.00% (0)	0.00% (0)	34	1.38	0	0
Projektbesprechung	14.71% (5)	8.82% (3)	14.71% (5)	5.88% (2)	32.35% (11)	23.53% (8)	34	4.03	0	0
romantisches Date	5.88% (2)	11.76% (4)	5.88% (2)	11.76% (4)	26.47% (9)	38.24% (13)	34	4.56	0	0
in einer Bar unter Freunden	23.53% (8)	20.59% (7)	35.29% (12)	11.76% (4)	8.82% (3)	0.00% (0)	34	2.62	0	0
sitzend in einem Bus	47.06% (16)	38.24% (13)	8.82% (3)	5.88% (2)	0.00% (0)	0.00% (0)	34	1.74	0	0
stehend in einer überfüllten U-Bahn	30.30% (10)	36.36% (12)	15.15% (5)	12.12% (4)	3.03% (1)	3.03% (1)	33	2.30	0	1
beim Einkaufen vor einem Regal	29.41% (10)	35.29% (12)	23.53% (8)	8.82% (3)	2.94% (1)	0.00% (0)	34	2.21	0	0
an der Kasse beim Bezahlen	9.09% (3)	18.18% (6)	30.30% (10)	18.18% (6)	12.12% (4)	12.12% (4)	33	3.42	0	1
kurz nach dem Aufwachen	33.33% (11)	30.30% (10)	9.09% (3)	15.15% (5)	3.03% (1)	9.09% (3)	33	2.52	0	1
beim Sport (z.B. Joggen)	21.21% (7)	27.27% (9)	24.24% (8)	15.15% (5)	3.03% (1)	9.09% (3)	33	2.79	0	1
beim Fernsehen	50.00% (17)	29.41% (10)	14.71% (5)	2.94% (1)	0.00% (0)	2.94% (1)	34	1.82	0	0

N = 34 | n = 34 | SYS-MISSING = 0

* A = N/A B = UNGÜLTIG (FEHLEND)

Frage: Wie störend würden Sie eine **MEHRMALIGE** Benachrichtigung zu einer passiven Störung in folgenden Situationen betrachten?

Bei mehrmaligen Benachrichtigungen werden Sie in bestimmten Intervallen benachrichtigt

	0 - NICHT STÖREND (1)	1 (2)	2 (3)	3 (4)	4 (5)	5 - SEHR STÖREND (6)	GESAMT	MITTELWERT	MISSING*	
									A	B
allein zu Hause	30.30% (10)	30.30% (10)	18.18% (6)	18.18% (6)	0.00% (0)	3.03% (1)	33	2.36	0	1
Projektbesprechung	2.94% (1)	2.94% (1)	0.00% (0)	8.82% (3)	29.41% (10)	55.88% (19)	34	5.26	0	0
romantisches Date	5.88% (2)	0.00% (0)	5.88% (2)	5.88% (2)	20.59% (7)	61.76% (21)	34	5.21	0	0
in einer Bar unter Freunden	2.94% (1)	11.76% (4)	20.59% (7)	23.53% (8)	29.41% (10)	11.76% (4)	34	4.00	0	0
sitzend in einem Bus	17.65% (6)	17.65% (6)	29.41% (10)	26.47% (9)	5.88% (2)	2.94% (1)	34	2.94	0	0
stehend in einer überfüllten U-Bahn	5.88% (2)	29.41% (10)	23.53% (8)	17.65% (6)	14.71% (5)	8.82% (3)	34	3.32	0	0
beim Einkaufen vor einem Regal	8.82% (3)	17.65% (6)	29.41% (10)	26.47% (9)	8.82% (3)	8.82% (3)	34	3.35	0	0
an der Kasse beim Bezahlen	3.03% (1)	6.06% (2)	21.21% (7)	12.12% (4)	30.30% (10)	27.27% (9)	33	4.42	0	1
kurz nach dem Aufwachen	11.76% (4)	23.53% (8)	17.65% (6)	14.71% (5)	14.71% (5)	17.65% (6)	34	3.50	0	0
beim Sport (z.B. Joggen)	8.82% (3)	8.82% (3)	14.71% (5)	41.18% (14)	11.76% (4)	14.71% (5)	34	3.82	0	0
beim Fernsehen	23.53% (8)	29.41% (10)	8.82% (3)	26.47% (9)	5.88% (2)	5.88% (2)	34	2.79	0	0

N = 34 | n = 34 | SYS-MISSING = 0

* A = N/A B = UNGÜLTIG (FEHLEND)

Frage: Wie würden Sie in der Mehrzahl der oben genannten Situationen mit den Benachrichtigungen umgehen?

	ANZAHL	PROZENT
die Benachrichtigungen nach der ersten Nachricht abschalten (1)	30	88.24%
kontinuierlich Nachrichten bekommen (2)	4	11.76%
GESAMT	34	
ungültig (fehlend)	0	

Mittelwert

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Im folgenden werden Ihnen Aussagen vorgelegt, die sich auf das intermodale Navigationssystem beziehen. Bewerten Sie nun diese Aussagen, inwiefern Sie eine positive, neutrale oder negative Haltung bei Ihnen auslösen. Sie haben z.B. die Aussage:

"Das intermodale Navigationssystem bietet immer die aktuellsten Informationen"

Sie können nun diese Aussage von:

sehr negativ (-5) - "Ich empfinde das als schlecht.", "Es stört mich sehr.", "Ich bin der Aussage gegenüber sehr negativ eingestellt."

über neutral (0) - "Ich empfinde diesen Punkt weder als positiv, noch als negativ.", "Ich sehe sowohl negatives, als auch positives."

bis sehr positiv (+5) "Ich empfinde das als gut/nützlich." "Das ist eine tolle Funktion" "Ich bin der Aussage gegenüber sehr positiv eingestellt."

bewerten.

	-5 [SEHR NEGATIV] (1)	-4 (2)	-3 (3)	-2 (4)	-1 (5)	0 [NEUTRAL] (6)	1 (7)	+2 (8)	+3 (9)	+4 (10)	+5 [SEHR POSITIV] (11)	GESAMT	MITTELWERT	MISSING*	
													A	B	
Das intermodale Navigationssystem sendet Benachrichtigungen, wann die Route gestartet werden soll.	0.00% (0)	0.00% (0)	0.00% (0)	2.94% (1)	0.00% (0)	14.71% (5)	11.76% (4)	20.59% (7)	20.59% (7)	8.82% (3)	20.59% (7)	34	8.47	0	0
Das intermodale Navigationssystem benutzt alle zur Verfügung stehenden Verkehrsmittel, um die Route zu berechnen.	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	2.94% (1)	8.82% (3)	2.94% (1)	5.88% (2)	17.65% (6)	14.71% (5)	47.06% (16)	34	9.59	0	0
Das intermodale Navigationssystem ändert bei einer abweichenden Startposition die Startzeit und weist den Nutzer darauf hin.	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	2.94% (1)	2.94% (1)	2.94% (1)	17.65% (6)	17.65% (6)	20.59% (7)	35.29% (12)	34	9.47	0	0
Das intermodale Navigationssystem erkennt mit welchem Verkehrsmittel man unterwegs ist.	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	2.94% (1)	2.94% (1)	11.76% (4)	17.65% (6)	26.47% (9)	38.24% (13)	34	9.76	0	0
Die angegebene Verspätung entsprach exakt der tatsächlichen Verspätung.	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	8.82% (3)	5.88% (2)	5.88% (2)	14.71% (5)	23.53% (8)	41.18% (14)	34	9.62	0	0
Die UBahn kam eine Minute früher als vom intermodalen Navigationssystem angegeben.	17.65% (6)	14.71% (5)	2.94% (1)	17.65% (6)	5.88% (2)	26.47% (9)	0.00% (0)	2.94% (1)	2.94% (1)	5.88% (2)	2.94% (1)	34	4.56	0	0
Die UBahn kam eine Minute später als vom intermodalen Navigationssystem angegeben.	0.00% (0)	0.00% (0)	3.03% (1)	6.06% (2)	12.12% (4)	39.39% (13)	18.18% (6)	15.15% (5)	3.03% (1)	0.00% (0)	3.03% (1)	33	6.39	0	1
Das intermodale Navigationssystem meldet, wenn auf der Route ungeplante Störungen auftreten, die Einfluss auf die Ankunftszeit haben.	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	3.03% (1)	0.00% (0)	18.18% (6)	24.24% (8)	21.21% (7)	33.33% (11)	33	9.61	0	1
Das intermodale Navigationssystem schlägt Alternativrouten vor.	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	2.94% (1)	11.76% (4)	29.41% (10)	20.59% (7)	35.29% (12)	34	9.74	0	0
Das intermodale Navigationssystem bietet alle wichtigen Informationen über die Route auf dem	0.00%	0.00%	0.00%	5.88%	0.00%	11.76%	5.88%	17.65%	20.59%	17.65%	20.59%	34	8.65	0	0

Sperbildschirm, ohne das Smartphone entsperren zu müssen.	(0)	(0)	(0)	(2)	(0)	(4)	(2)	(6)	(7)	(6)	(7)				
Das intermodale Navigationssystem sendet häufig Mitteilungen und Updates.	2.94% (1)	5.88% (2)	17.65% (6)	8.82% (3)	20.59% (7)	23.53% (8)	5.88% (2)	5.88% (2)	2.94% (1)	2.94% (1)	2.94% (1)	34	5.24	0	0
Das intermodale Navigationssystem sendet keine Mitteilungen und Updates.	5.88% (2)	8.82% (3)	8.82% (3)	23.53% (8)	17.65% (6)	20.59% (7)	2.94% (1)	2.94% (1)	5.88% (2)	2.94% (1)	0.00% (0)	34	4.82	0	0
Das intermodale Navigationssystem erkennt, dass man in das falsche öffentliche Verkehrsmittel gestiegen ist.	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	2.94% (1)	2.94% (1)	17.65% (6)	29.41% (10)	14.71% (5)	32.35% (11)	34	9.47	0	0
Das intermodale Navigationssystem erkennt, dass der Nutzer noch nicht losgegangen ist und informiert ihn sofort	0.00% (0)	0.00% (0)	0.00% (0)	5.88% (2)	0.00% (0)	17.65% (6)	5.88% (2)	11.76% (4)	20.59% (7)	11.76% (4)	26.47% (9)	34	8.59	0	0
Nachdem die Route wie vom intermodalen Navigationssystem vorgegeben angetreten wurde, wurde der Bus rechtzeitig erreicht.	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)	8.82% (3)	17.65% (6)	20.59% (7)	11.76% (4)	41.18% (14)	34	9.59	0	0
Nachdem die Route wie vom intermodalen Navigationssystem vorgegeben angetreten wurde, wurde der Bus verpasst.	47.06% (16)	29.41% (10)	0.00% (0)	5.88% (2)	2.94% (1)	5.88% (2)	2.94% (1)	0.00% (0)	2.94% (1)	2.94% (1)	0.00% (0)	34	2.56	0	0

N = 34 | n = 34 | SYS-MISSING = 0

* A = N/A B = UNGÜLTIG (FEHLEND)

Frage: Ein intermodales Navigationssystem benötigt standortbezogene Daten(z.B. GPS), um im vollen Funktionsumfang funktionieren zu können. Im Folgenden werden Fragen gestellt, die sich auf das GPS beziehen. Diese Fragen beantworten Sie bitte mit Ihrer persönlichen Meinung im Bereich von "stört mich gar nicht" bis "stört mich sehr".

	0 [STÖRT MICH GAR NICHT] (1)	1 (2)	2 (3)	3 (4)	4 (5)	5 [STÖRT MICH SEHR] (6)	GESAMT	MITTELWERT	MIS A
Inwieweit stört es Sie, dass Sie das GPS aktiviert haben müssen, damit das intermodale Navigationssystem korrekt funktioniert?	35.29% (12)	23.53% (8)	5.88% (2)	26.47% (9)	5.88% (2)	2.94% (1)	34	2.53	0
Stört Sie ein erhöhter Akkuverbrauch in Folge der GPS Aktivierung?	2.94% (1)	11.76% (4)	26.47% (9)	20.59% (7)	14.71% (5)	23.53% (8)	34	4.03	0

N = 34 | n = 34 | SYS-MISSING = 0

* A = N/A B = UNGÜLTIG (FEHLEND)

Frage: Haben Sie bereits Erfahrung mit Widgets, die auf dem Sperbildschirm angezeigt werden? z.B. Spotify

	ANZAHL	PROZENT
Ja (1)	22	64.71%
Nein (2)	12	35.29%
GESAMT	34	
ungültig (fehlend)	0	
Mittelwert	1.35	

N = 34 | n = 34 | SYS-MISSING = 0

Frage: Bitte ordnen Sie die folgenden Themen einer Kategorie zu, die auf dem Widget, das auf dem Lockscreen platziert wird, angezeigt werden soll. (von unwichtig bis sehr wichtig)

	0 - UNWICHTIG (1)	1 (2)	2 (3)	3 (4)	4 (5)	5 - WICHTIG (6)	GESAMT MIT
Klassifizierung – um was für eine Störung es sich handelt (z.B. Verspätung der U-Bahn)	2.94% (1)	0.00% (0)	8.82% (3)	23.53% (8)	35.29% (12)	29.41% (10)	34

Lösungen – Benutzer wird über mögliche Aktionsmöglichkeiten hingewiesen (z.B. Alternativroute nehmen (5 min Verspätung) oder warten(15 Minuten Verspätung))	2.94% (1)	5.88% (2)	5.88% (2)	17.65% (6)	29.41% (10)	38.24% (13)	34
Konsequenzen – Auswirkungen, wenn die Route nicht eingehalten wird (z.B. S-Bahn verspätet sich – Anschluss U-Bahn wird verpasst – Ankunftszeit 10 Minuten später)	5.88% (2)	11.76% (4)	20.59% (7)	23.53% (8)	23.53% (8)	14.71% (5)	34
Verspätungsanzeige (U2 hat etwa 5 Minuten Verspätung)	2.94% (1)	2.94% (1)	11.76% (4)	29.41% (10)	29.41% (10)	23.53% (8)	34
alle Stationen der Route sollen angezeigt werden	44.12% (15)	35.29% (12)	14.71% (5)	2.94% (1)	2.94% (1)	0.00% (0)	34
die nächste Station, die erreicht wird soll angezeigt werden	8.82% (3)	20.59% (7)	23.53% (8)	8.82% (3)	32.35% (11)	5.88% (2)	34

N = 34 | n = 34 | SYS-MISSING = 0

* A = N/A B = UT

Frage: Bitte ordnen Sie die folgenden Themen einer Kategorie zu, die in der Notification angezeigt werden soll.
(von unwichtig bis sehr wichtig)

	0 - UNWICHTIG (1)	1 (2)	2 (3)	3 (4)	4 (5)	5 - WICHTIG (6)	GESAMT MIT
Klassifizierung – um was für eine Störung es sich handelt (z.B. Verspätung der U-Bahn)	2.94% (1)	5.88% (2)	8.82% (3)	23.53% (8)	23.53% (8)	35.29% (12)	34
Lösungen – Benutzer wird über mögliche Aktionsmöglichkeiten hingewiesen (z.B. Alternativroute nehmen (5 min Verspätung) oder warten(15 Minuten Verspätung))	5.88% (2)	5.88% (2)	14.71% (5)	14.71% (5)	29.41% (10)	29.41% (10)	34
Konsequenzen – Auswirkungen, wenn die Route nicht eingehalten wird (z.B. S-Bahn verspätet sich – Anschluss U-Bahn wird verpasst – Ankunftszeit 10 Minuten später)	5.88% (2)	14.71% (5)	11.76% (4)	26.47% (9)	20.59% (7)	20.59% (7)	34
Verspätungsanzeige (U2 hat etwa 5 Minuten Verspätung)	5.88% (2)	0.00% (0)	26.47% (9)	26.47% (9)	20.59% (7)	20.59% (7)	34
alle Stationen der Route sollen angezeigt werden	50.00% (17)	29.41% (10)	14.71% (5)	2.94% (1)	2.94% (1)	0.00% (0)	34
die nächste Station, die erreicht wird soll angezeigt werden	20.59% (7)	14.71% (5)	14.71% (5)	17.65% (6)	26.47% (9)	5.88% (2)	34

N = 34 | n = 34 | SYS-MISSING = 0

* A = N/A B = UT

Tagebuch

Datum:

1. *Wie oft haben Sie das intermodale Navigationssystem heute benutzt?*
2. *Die verwendeten Transportmittel wurden von dem intNAV korrekt erkannt.*
3. *Die aufgetreten aktiven Abweichungen wurden von dem intNAV korrekt erkannt.*
4. *Die angezeigten Empfehlungen bei erkannten aktiven Störungen waren hilfreich.*
5. *Die Störungen wurden auf eine angemessene Art und Weise kommuniziert.*
6. *Ich fühlte mich auf meinen Routen durch das intNAV sehr gut unterstützt.*
7. *Die dargestellten Informationen auf dem intNAV beurteile ich als hilfreich.*

Abschlussinterview

1. *Bitte lassen Sie die vergangene Woche nochmals Revue passieren – für welche Situationen haben Sie das intNAV verwendet?*
2. *In welchen Situationen war die Applikation am hilfreichsten und bei welchen am überflüssigsten?*
3. *Wurden Ihre Erwartungen an die Applikation enttäuscht, erfüllt oder übertroffen?*
4. *Nennen Sie 3 Punkte, die Ihnen positiv im Gedächtnis geblieben sind.*
5. *Nennen Sie Punkte, die Ihnen negativ im Gedächtnis geblieben sind.*
6. *Welche Features waren hilfreich/ welche nicht?*
7. *Was haben Sie an der Applikation am meisten vermisst (im Vergleich mit anderen Navigationssystemen)?*

