

EINE MOBILITÄTSPLATTFORM FÜR DEN BEDARFSORIENTIERTEN VERKEHR IM LÄNDLICHEN RAUM

Eric Neidhardt¹

¹ Berlin, DLR Institut für Verkehrssystemtechnik, eric.neidhardt@dlr.de

Keywords: Bedarfsorientierter Verkehr, HubChain, Reallabor, Öffentlicher Verkehr

ABSTRACT

Mobilität ist für viele Menschen essentiell um ihre täglichen Bedürfnisse befriedigen zu können. Menschen müssen zur Arbeit gelangen, Einkaufen gehen und Arztbesuche tätigen. Besonders ältere Mitmenschen oder Personen mit Einschränkungen sind für ihre Mobilität auf öffentliche Verkehrsmittel angewiesen. Das ist besonders im dünn besiedelten, ländlichen Raum problematisch. In diesem Bereich ist es selten kosteneffizient einen attraktiven und deckenden Öffentlichen Verkehr anzubieten. Durch geringe Abdeckung und eingeschränkte Nutzbarkeit ist der Öffentliche Verkehr oft nicht ausreichend um die Bedürfnisse der darauf angewiesenen Menschen zu befrieden. Dadurch sinkt die Nutzung des Öffentlichen Verkehrs zusätzlich. In manchen Regionen organisieren sich die Menschen deswegen selbst und betreiben ehrenamtliche Mobilitätsdienste. Diese Bürgerbusse operieren teilweise bedarfsorientiert. Fahrgäste buchen den Bus telefonisch im Voraus und die ehrenamtlichen Fahrer planen ihre Route händisch aus diesen Anfragen. Im Projekt HubChain haben wir eine Mobilitätsplattform entwickelt um beliebige, bedarfsorientierte Mobilitätsangebote miteinander zu verknüpfen. Der Fokus in der Entwicklung wurde dabei auf die bedarfsorientierten Bürgerbusse im Elde-Quellgebiet gelegt. Die HubChain Plattform unterstützt die Planung über verschiedenen Bedingebiete und Bedarfsorientierte Mobilitätsangebote sowie die Anbindung an den konventionellen, liniengebundenen Öffentlichen Verkehr. Die zugrundeliegende Technologie der HubChain Plattform ist das DLR KeepMoving Serviceframework. Die einzelnen Komponenten der Plattform sind modular und flexibel aufgebaut. Über die HubChain Plattform können Nutzer intermodale Routen planen, die sowohl den Linienverkehr, als auch die angeschlossenen bedarfsorientierten Mobilitätsangebote unterstützen. Diese Routen können direkt über ein Web API oder eine App gebucht werden. Ziel der HubChain Plattform ist es vor allem die Attraktivität und den Nutzen des Öffentlichen Verkehrs im ländlichen Raum zu steigern.

1 EINLEITUNG

Mit herkömmlichen, liniengebundenen Systemen für den öffentlichen Verkehr ist es oft schwierig, ein ausreichendes Maß an Flexibilität zu erreichen. Diese starren Ansätze sind in Großstädten gut anwendbar, da dort eine große und ständige Nachfrage besteht. Mit Anpassungen an der Taktung und der Größe der eingesetzten Fahrzeuge lassen sich Bedarfsspitzen und Zeiten mit niedrigem Bedarf abfedern.

In ländlichen und vorstädtischen Gebieten ist der Einsatz von liniengebundenen Systemen schwieriger, weil es teuer und aufwendig ist eine deckende Versorgung zu gewährleisten. Infolgedessen werden nur wenige Fahrten angeboten. Dies führt zu einem unattraktiven System, das noch weniger von den Nutzern genutzt wird [2].

Bedarfsorientierten Verkehrsansätzen sind ein Versuch eine flexiblere und damit kosteneffizientere Alternative anzubieten. Leerfahrten und unnötige Umwege sollen vermieden und gleichzeitig die Reisezeiten verkürzt werden. Ziel ist es, ein attraktives Mobilitätsangebot zu schaffen. Im Fokus stehen dabei vor allem Menschen ohne eigenes Fahrzeug. Bei Bedarfsorientierten Mobilitätsangeboten ist es wichtig darauf zu achten, dass sie nicht in Konkurrenz zu den bestehenden öffentlichen Verkehrsmitteln stehen, weil das die Nachhaltigkeit reduzieren würde. Durch eine Kopplung von bedarfsorientierten Diensten und dem liniengebundenen öffentlichen Verkehr soll die Attraktivität des öffentlichen Verkehrs gesteigert werden. Die Kopplung kann in Form einer Anschlussgarantie oder einer gesamtheitlichen Routenplanung erfolgen.

Das Anwendungsgebiet unserer Mobilitätsplattform war das Elde-Quellgebiet in Mecklenburg-Vorpommern. Das Gebiet ist ein sehr ländliches Gebiet mit spärlicher Besiedlung und tendenziell älterer Bevölkerung. Die HubChain Plattform wird in einem Reallabor zum Einsatz kommen und die in dem Anwendungsgebiet vorhandenen Angebote für Bürgerbusse miteinander verknüpfen und mit dem liniengebundenen Verkehr koppeln. Der Testbetrieb erfolgt in Form eines Reallabor. Ein Reallabor ist eine als Kooperation zwischen Zivilgesellschaft und Wissenschaft setzt die Forschung in einem praxisnahen Umfeld in den Vordergrund [1].

Aufgrund der SARS-CoV-2 Pandemie konnten praktische Einsätze nicht durchgeführt werden. Die praktische Erprobung wird Bestandteil zukünftiger Projekte sein.

2 HINTERGRUND UND BISHERIGE ARBEITEN

Bedarfsorientierte Mobilitätskonzepte existieren bereits länger. Viele dieser Systeme liegen zwischen dem Individualverkehr und einem regelmäßigen, linienbasierten Busverkehr. In Großbritannien entstanden einige dieser Systeme bereits im Jahr 1970 [3]. Aktuellere Beispiele, wie zum Beispiel der Verkehrsdienstleister Uber, haben eher den Charakter eines Ruftaxis. Diese Mobilitätsangebote sind deswegen auch häufig auf den städtischen Raum konzentriert und in ländlichen Regionen nur eingeschränkt präsent.

Einen ähnlichen Ansatz wie mit der HubChain Plattform haben wir im vorherigen Projekt, Reallabor Schorndorf, untersucht. Die Forschungsfrage in Schorndorf war aber eine andere. Im Reallabor Schorndorf war unser Ziel, die Nachhaltigkeit des bestehenden ÖPNV durch Hinzufügen eines nachfrageorientierten Ansatzes zu verbessern. In Zeiten geringer Nachfrage wurde der reguläre, linienbasierte Busverkehr durch einen bedarfsorientierten Busverkehr ersetzt. Die primäre Herausforderung des Reallabor Schorndorfs war die Akzeptanz der Kunden und die rechtlichen und administrativen Herausforderungen. Zwar konnte das System die Zahl der Leerfahrten reduzieren, aber auch die Fahrgastzahlen waren geringer. Einige Fahrgäste lehnten das System aus verschiedenen Gründen ab [5].

3 BEDIENGEBIET

Das Einsatzgebiet für die Erprobung der HubChain Plattform war das Elde-Quellgebiet in Mecklenburg-Vorpommern. Das Elde-Quellgebiet ist eine dünn besiedelte, ländliche Region mit hohem Durchschnittsalter. Das Bediengebiet umfasst die Gemeinden Röbel, Bollewick, Ludorf, Solzow und Dambeck. Es stehen nur wenige Öffentliche Verkehrsmittel zur Verfügung. Dadurch ist es für ältere Menschen teilweise äußerst schwierig, ihre täglichen Bedürfnisse zu erfüllen. Meist müssen sich ältere Einwohner auf die Hilfe ihrer Kinder oder Nachbarn verlassen. Der Besitz eines eigenen Autos ist daher für viele Einwohner besonders wichtig.

Abbildung 1 zeigt das Einsatzgebiet rund um die Stadt Röbel. Die Organisation und der Betrieb der bedarfsorientierten Shuttles wurde durch das Institut KOMOB durchgeführt. Die Shuttles wurden eingesetzt, um mehrere kleine Ortschaften an den liniengebundenen ÖPNV der örtlichen Verkehrsgesellschaft (MVVG) anzubinden. Zu Beginn des Projekts wurde die Routenplanung für diese Flotten manuell von den ehrenamtlichen Fahrern durchgeführt. Später wurden diese Angebote an die HubChain-Plattform angeschlossen. Die Plattform übernahm die Aufgabe der Fahrzeug Disposition [7]. Aufgrund der einsetzenden SARS-CoV-2 Pandemie konnte ein umfassender Test unter realen Bedingungen nicht durchgeführt werden.

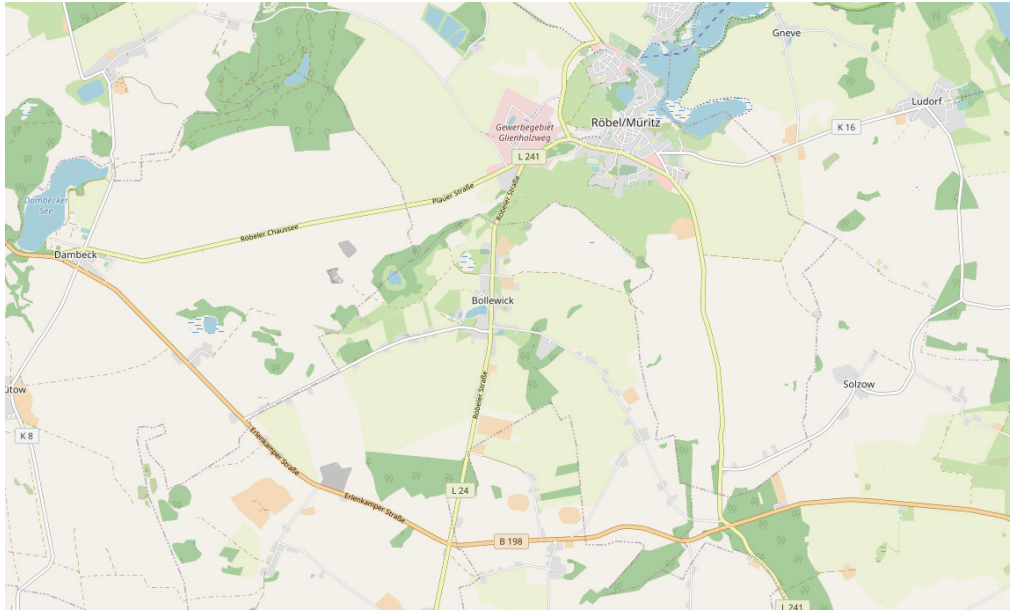


Abbildung 1: Elde-Quellgebiet (© OpenStreetMap und Mitwirkende, CC-BY-SA).

4 TECHNISCHE UMSETZUNG

In diesem Abschnitt werden die technische Infrastruktur und die Implementierung der Backend-Dienste für die HubChain-Plattform beschrieben. Der Fokus der Beschreibung liegt dabei auf den Schnittstellen und der Nutzerinteraktion. Die Abbildungen 2 bis 4 in diesem Abschnitt werden auch in [13] verwendet.

Die Implementierung der HubChain-Plattform basiert auf dem KeepMoving-Framework. KeepMoving ist ein Service-Framework zur Erfassung, Verarbeitung und Nutzung von Verkehrsdaten. Es stellt unter anderem Dienste für intermodales Routing und Netzanpassung zur Verfügung. Verkehrsdaten können aus verschiedenen Quellen bezogen werden, zum Beispiel aus Schleifendetektor-Daten oder Floating-Car-Daten. Diese Daten können in die Routing-Dienste eingebunden werden, sodass die aktuelle Verkehrslage berücksichtigt wird. KeepMoving ist in der Lage, auf OpenStreetMap-Daten zu arbeiten [6].

Die Implementierung der HubChain-Plattform erfolgte als serviceorientierte Architektur. Sie ist flexibel und kompatibel zu KeepMoving. Eine serviceorientierte Implementierung erleichtert die Anpassung der Software an sich ändernden Anforderungen. Außerdem können auf diese Weise einzelne Komponenten für kommende Projekte ausgetauscht oder wiederverwendet werden. Alle Dienste wurden in der Programmiersprache Java implementiert. Die Anbindung der Dienste untereinander erfolgte über einen Apache ActiveMQ Messaging Broker [8]. ActiveMQ implementiert die Java Message Service (JMS) API und JMS ObjectMessages werden von den verschiedenen Diensten verwendet. Der externe Zugriff wurde über SOAP und RESTful Webservices realisiert.

Das Kartenmaterial wurde von OpenStreetMap bezogen. Das SUMO-Tool `osmGet.py` wird verwendet, um OpenStreetMap-Daten im OSM-XML-Format herunterzuladen. Ein weiteres SUMO-Werkzeug (`netconvert tool`) wird verwendet, um die heruntergeladenen OSM-Daten in unser internes Format zu konvertieren [9]. Diese Daten werden in der Netzdatenbank gespeichert. Es werden sowohl PostGIS als auch Oracle Spatial and Graph unterstützt. Die importierten Daten können später manuell verfeinert werden, indem andere Informationsquellen, wie z. B. Daten von lokalen Behörden oder der Gemeinde, eingebunden werden. Dies ist besonders wichtig, wenn die importierten Kartendaten nicht vollständig oder veraltet sind.

Für die intermodale Routenberechnung werden Fahrplaninformationen des lokalen Mobilitätsanbieters (MVVG) berücksichtigt. Die Daten werden im TRIAS-Format importiert. Das TRIAS-Format (Travellers Realtime Information Advisory Standard) wurde vom Verband Deutscher Verkehrsunternehmen (VDV) entwickelt und wird von vielen deutschen Mobilitätsanbietern

unterstützt [10].

Abbildung 2 zeigt einen vereinfachten Überblick über das Gesamtsystem. Wie zuvor beschrieben, ist das System als eine Reihe von mehreren Diensten implementiert. Diese Dienste kommunizieren über einen gemeinsamen Message Broker, der in dieser Abbildung zur besseren Übersicht nicht dargestellt ist. Anbieter von Mobilitätsangeboten können sich mit Hilfe des Registrierungsdienstes am System registrieren. Sie können die Randbedingungen ihres Dienstes eingeben, wie Einsatzgebiet, Einsatzzeit und Fahrzeugkapazität. Diese Daten werden in der Flottendatenbank gespeichert.

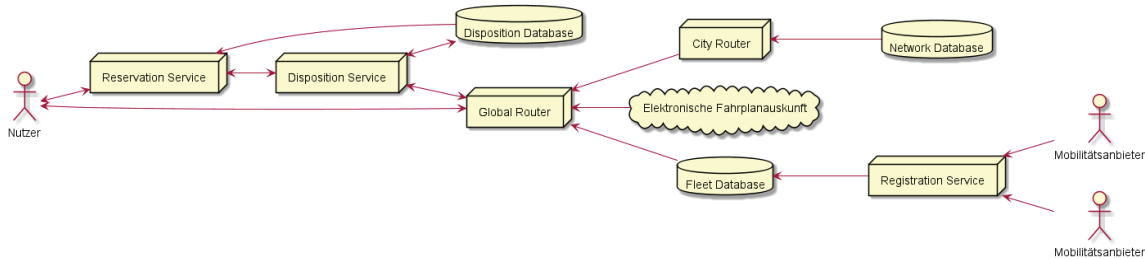


Abbildung 2: HubChain Plattform.

Der City-Router ist die primäre Routing-Engine. Er unterstützt monomodale Routen für Autos, Busse, Fußgänger und mehr. Intern gibt es verschiedene Instanzen des Routers, basierend auf der Transportart. Der City-Router wird vom sogenannten Global-Router aufgerufen, der als Broker für Routing-Anfragen fungiert. Der Global-Router ist für die Berechnung der intermodalen Routen zuständig. Er berücksichtigt dabei Fahrplaninformationen des lokalen Mobilitätsanbieters und alle angemeldeten bedarfsorientierten Dienste. Eine besonders wichtige Komponente ist der Dispositionsdienst. Der Dispositionsdienst ist für die Erstellung der Verkehrspläne für die bedarfsorientierten Shuttles auf Basis der Benutzeranfragen zuständig. Der Dispositionsdienst kommuniziert mit dem Global-Router, um Routen und Fahrzeiten für die Shuttles zu erhalten. Der Reservierungsdienst ist hauptsächlich für die Verwaltung von Buchungen und Reservierungen zuständig.

Die verschiedenen APIs aus Nutzersicht sind in Abbildung 3 dargestellt. Bei den APIs für Routing und Buchungen handelt sich um die gleichen Schnittstellen wie in Abbildung 1. Die Buchungs-API ist für die Verwaltung von Buchungen zuständig. Die Anwendungsfälle sind Reservieren, Stornieren und Buchen von bedarfsorientierten Fahrten. Die Routing-API ist für die Bereitstellung von Routen für den Benutzer, basierend auf seiner Anfrage, verantwortlich. Diese Routen können bedarfsorientierte Dienste enthalten, müssen es aber nicht. Wenn eine Anfrage mit konventionellen öffentlichen Verkehrsmitteln befriedigt werden kann, wird dieses Ergebnis aus Gründen der Nachhaltigkeit bevorzugt. Die Journey-API wird verwendet, um Informationen über geplante Routen bereitzustellen. Sie dient hauptsächlich dazu, die Fahrt für den Benutzer zu visualisieren. In der eigentlichen Implementierung werden alle APIs über eine gemeinsame Schnittstelle, den ServiceHub, gekapselt. Diese Kapselung ist in der Darstellung, zur besseren Übersicht, nicht dargestellt.

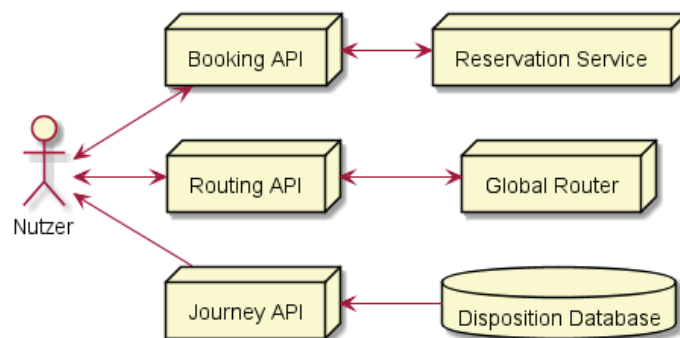


Abbildung 3: Schnittstellen aus Nutzersicht.

Der Ablauf der primären Anwendungsfälle für die Interaktion mit dem bedarfsorientierten Angebot aus der Sicht des Benutzers sind in Abbildung 4 als Sequenz-Diagramm dargestellt. Diese Anwendungsfälle umfassen die Planung und Reservierung einer intermodalen Fahrt mit bedarfsorientiertem Anteil, die Buchung einer reservierten Fahrt sowie die Stornierung. Wenn die Route des Nutzers einen bedarfsorientierten Anteil hat, dann muss dieser Anteil der Fahrt reserviert und gebucht werden. Wir haben uns für einen zweistufigen Buchungsprozess entschieden, bestehend aus einer Planungs- bzw. Reservierungsphase und der eigentlichen Buchung. Nachdem der Benutzer eine Route mit einer On-Demand-Komponente angefordert hat, führt der Dispositionsdienst die notwendigen Berechnungen für die Fahrt durch. Die Disposition versucht, die Anfrage mit bereits geplanten Fahrten zusammenzuführen. Wenn dies nicht möglich ist wird eine neue Fahrt erstellt. Die Reiseinformationen werden an den Benutzer zurückgegeben. In der realen Implementierung wird eine Liste mit mehreren Ergebnissen (sofern verfügbar) an den Benutzer zurückgegeben werden, um ihm Optionen zu bieten. Darüber hinaus könnte es möglich sein, dass die Benutzeranforderung nur durch die Verwendung von klassischen, linienbasierten Diensten erfüllt werden kann. In diesem Fall ist keine Reservierung erforderlich und der Benutzer wird darüber informiert, dass er die normalen öffentlichen Verkehrsmittel nutzen kann. Wenn die berechnete Route eine bedarfsorientierte Komponente enthält, ist eine Reservierung erforderlich. In diesem Stadium wird die berechnete Route bereits in die Dispositionsdatenbank aufgenommen. Sie ist allerdings als vorläufig gekennzeichnet. Es wird eine generierte Buchungsnummer zusammen mit einer Anfrage-ID (`request_id`) zurückgegeben. Diese IDs werden bei weiteren Anfragen zur Bestätigung oder Stornierung der Fahrt verwendet. Der Benutzer hat 5 Minuten Zeit um seine Fahrt zu bestätigen, ansonsten wird die vorläufige Route wieder entfernt. Da die vorläufige Route bereits in weiteren Routenberechnungen berücksichtigt wird, kann sie nicht durch eine weitere Anfrage eines anderen Nutzers ungültig gemacht werden. Dies hat den Nachteil, dass bei vielen Anfragen in kurzer Zeit und begrenzten Möglichkeiten, die Disposition schnell mit vorläufigen Routen überladen sein kann. Da wir uns jedoch auf ländliche Gebiete konzentrieren, war dies bei unseren Tests kein Problem.

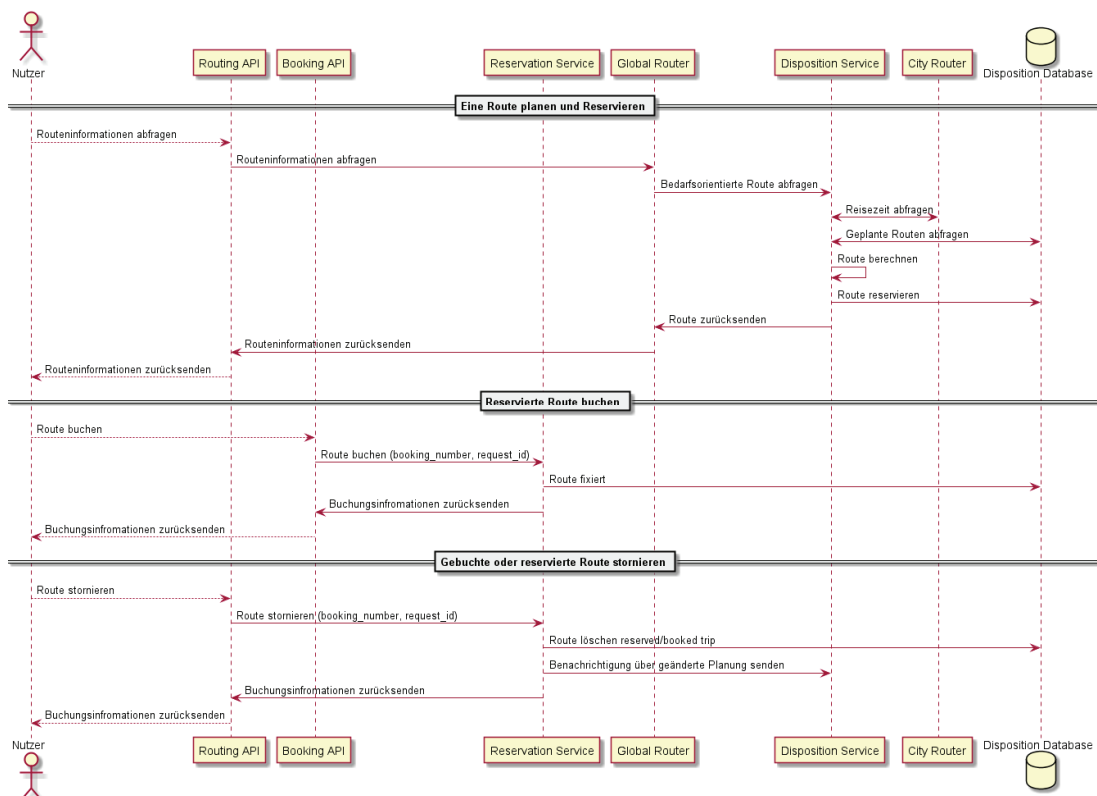


Abbildung 4: Buchen, Reservieren, Stornieren.

Die Buchung und Bestätigung einer vorläufigen Route ist einfach, weil die Route bereits vollständig in die Datenbank eingepflegt ist. Der Reservierungsdienst überprüft die bereitgestellten Informationen (Buchungsnummer und Anfrage-ID) und aktualisiert die Kennzeichnung der Route entsprechend. Der Prozess für die Stornierung ist ähnlich und funktioniert für vorläufige Routen und gebuchte Routen auf die gleiche Weise. Die Stornierung einer Anfrage macht jedoch eine Neuberechnung der betroffenen Routen erforderlich, weil bestimmte geplante Haltestellen möglicherweise nicht mehr notwendig sind. Der Reservierungsdienst sendet eine Benachrichtigung an den Dispositionsdienst. Die Disposition führt darauf eine Neuberechnung aller betroffenen Routen durch. Nutzer deren Reisedaten sich dadurch ändern, werden über die geänderten Reisedaten zu informiert.

4.1 Disposition Algorithmus

Die Disposition eines bedarfsorientierten Shuttles auf Basis von Benutzeranfragen ist ein Dial-a-Ride-Problem. Das Lösen eines Dial-a-Ride-Problems kann mit mehreren Zielen erfolgen. Oft ist es gewünscht, ein Maximum an Anfragen mit einer gegebenen Flottengröße zu bedienen. Der Dispositionsdienst der HubChain-Plattform wurde mithilfe eines Ameisenalgorithmus implementiert. Bei diesem Ansatz arbeiten Agenten zusammen, um das Travelling-Salesman-Problem zu lösen [12]. Details zur Implementierung des Algorithmus wie er im Dispositionsdienst verwendet wird sind in [11] beschrieben.

4.2 Nutzeranfragen zusammenfassen und Randbedingungen

Ähnliche Nutzeranfragen werden zur Erhöhung der Nachhaltigkeit zusammengefasst, damit die Auslastung der bedarfsorientierten Mobilitätsangebote möglichst hoch ist. Dieses Clustering findet nur bei Routen mit einer bedarfsorientierten Komponente Anwendung. Fahrtabschnitte mit dem Liniengebundenen Öffentlichen Verkehr sind davon nicht beeinflusst. Für die Zusammenfassung der Nutzeranfragen ist die Berücksichtigung mehrerer Randbedingungen erforderlich. Randbedingungen haben einen starken Einfluss auf das Bündelungsverhalten, die Kapazität und die Nachhaltigkeit des Systems. Sie beeinflussen aber auch die Attraktivität des Systems für den Benutzer. Die Planung weniger, aber längerer Routen kann durch das Zusammenführen von Buchungsanfragen erreicht werden. Weniger Routen zu planen, ist nachhaltiger. Allerdings verringert dieses Verhalten auch die Attraktivität des Systems aus Sicht des Benutzers, weil Laufwege und Reisezeiten steigen.

Die Randbedingungen die von der Disposition berücksichtigt werden müssen unterteilen sich in 3 Bereiche. Nutzerzentrierte, technische und organisatorische Randbedingungen.

Organisatorische Randbedingungen umfassen zum Beispiel das Bediengebiet und die Einsatzzeiten. Technische Randbedingungen beziehen sich auf das verwendete Fahrzeug. Hierzu gehören unter anderem die Kapazität und die Reichweite. Sowohl die organisatorischen, als auch die technischen Randbedingungen werden vom Anbieter des Mobilitätsangebots festgelegt und können von der HubChain Plattform nicht beeinflusst werden.

Nutzerzentrierte Randbedingungen haben den größten Einfluss auf das Clustering-Verhalten des Dispositionsdienstes. Wenn Anfragen zusammengefasst werden kann der tatsächlich geplante Abholort sowie die Abfahrts- oder Ankunftszeit von der ursprünglichen Benutzeranfrage abweichen. Diesen Änderungen sind jedoch Grenzen gesetzt, da das System sonst für den Benutzer zu unattraktiv wäre. Um die möglichen Grenzen abschätzen zu können, sind zwei Faktoren im Benutzerverhalten besonders wichtig. Der erste Faktor ist die maximale Entfernung, die ein Benutzer bereit ist zum Abholort zu laufen. Der zweite Faktor ist die maximale Abweichung von seiner gewünschten Abfahrts- oder Ankunftszeit, die er bereit ist zu akzeptieren. Die benutzerzentrierten Randbedingungen werden vom Anbieter der HubChain-Plattform festgelegt. Um möglichst gute Werte zu bestimmen ist die Einbeziehung der Kunden und Benutzern erforderlich,

9 ZUSAMMENFASSUNG UND AUSBLICK

Insgesamt verlief die Implementierung der HubChain-Plattform in dem ausgewählten ländlichen Testgebiet erfolgreich. Das Projekt lieferte einige wertvolle Anregungen für die weitere Entwicklung.

Allerdings gab es auch Probleme. Leider konnten aufgrund der Auswirkungen der SARS-CoV-2-Pandemie ein realer Testbetrieb nicht so umfangreich wie von uns gewünscht durchgeführt werden. Weitere Tests und der Einsatz in zukünftigen Projekten sind notwendig, um die Software zu bewerten.

Derzeit berücksichtigt der Dispositionsdienst nicht die aktuelle, tatsächliche Position der Fahrzeuge. Es wird davon ausgegangen, dass die disponierten Fahrzeuge wie geplant fahren. Da das Einsatzgebiet sehr ländlich geprägt ist, stellt dies in der Regel kein Problem dar. Staus sind selten und die Routenplanung und Buchung erfolgt mit ausreichend zeitlichem Vorlauf. Aufgrund der geringen Nutzerzahlen ist es leicht die Nutzer im Falle einer Anomalie zu informieren. Trotzdem müssen bei der Routenplanung Puffer eingeplant werden, um unvorhergesehene Verspätungen zu kompensieren. Sollte eine Ausnahme auftreten, ist ein manueller Eingriff erforderlich.

Der Flottenregistrierungsdienst, um neue On-Demand-Flotten zum System hinzuzufügen, ist ebenfalls noch nicht implementiert. Das Hinzufügen neuer Mobilitätsangebote und das Ändern der Randbedingungen bestehender Flotten erfordert derzeit manuelle Änderungen in der Datenbank, was umständlich und fehleranfällig ist.

DANKSAGUNG

Die HubChain-Plattform wurde im Projekt HubChain entwickelt, das vom BMVI finanziert wurde.



Bundesministerium
für Wirtschaft
und Energie

REFERENCES

- [1] U. Schneidewind and K. Borschert, „Wissenschaft für Nachhaltigkeit - Herausforderung und Chance für das baden-württembergische Wissenschaftssystem, 2013.
- [2] J. Berg and J. Ihlström, “The Importance of Public Transport for Mobility and Everyday Activities among Rural Residents, *Open Access Journal*, vol. 8(2), 2019, pp. 1-13.
- [3] P. Bakker, “Large scale Demand Responsive Transit Systems - A Local Suburban Transport Solution for the Next Millennium?, *Proceedings of European Transport Conference, Stream: Public Transport Planning and Management*, 1999.
- [4] M. P. Enoch, S. Potter, G. P. Parkhurst, M. Smith, *Why do Demand Responsive Transport Systems fail?*” in *85th annual meeting of the TRB*, 2006.
- [5] E. Neidhardt and A. Sauerländer-Biebl, Provisioning a demand-orientated bus system for public transportation, 2018.
- [6] L. C. T. Tcheumadjeu, S. Ruppe, “Traffic Data Platform based on the Service Oriented Architecture (SOA), 2010.
- [7] Stadtwerke Osnabrück AG, Hub Chain project, <https://www.hubchain.de/aktuelles/>, 2011.
- [8] Apache ActiveMQ, <http://activemq.apache.org>, 2019.
- [9] SUMO, Simulation of Urban Mobility, <https://sumo.dlr.de/index.html>, 2019.
- [10] Internet Protokoll basierte Kommunikationsdienste im ÖV - Das Projekt IP-KOM-ÖV, <https://www.vdv.de/ip-kom-oev.aspx>, 2020.
- [11] Q. Tang and E. Neidhardt, Simulation-based method of a dynamical on-demand transportation problem, *Proceedings of WTC 2019, 3rd World Transport Convention*, 2019.
- [12] Marco Dorigo and L. M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation* vol.1, 1997.
- [13] E. Neidhardt, Implementing a Mobility Platform to Connect Hubs in Rural Areas, *unpublished*, 2021.