

国土交通省 Project LINKS × 地域交通DX推進プロジェクト「COMmmONS(コモンズ)」

2025年度 GTFS-Flex及びGTFS-Ondemandの技術実証プロジェクト

GTFS-Flex/Ondemand活用システム 技術検証レポート

Technical Report on GTFS-Flex/Ondemand Utilization Systems



技術検証レポートについて

国土交通省 Project LINKS × 地域交通DX推進プロジェクト「COMmmONS(コモンズ)」
2025年度 GTFSS-Flex及びGTFSS-Ondemandの技術実証プロジェクト

GTFSS-Flex/Ondemand活用システム 技術検証レポート

Technical Report on GTFSS-Flex/Ondemand Utilization Systems



国土交通省 総合政策局 情報政策課/モビリティサービス推進課

(技術資料リンク) No.009

- 技術検証レポートは、Project LINKS及びCOMmmONSにおける技術開発成果を広く社会一般に知見として提供するため、プロジェクトの有用性、実現性、課題等を整理したドキュメントです。
- 具体的には以下の役割を果たすものとして作成しています。
 - プロジェクトにおいては、地域交通における課題の設定とそれらを解決するためのデジタル技術活用のベストプラクティスを開発します。
 - 技術検証レポートは、プロジェクトの成果を社会の共通の財産とするため、関連技術の開発や研究、企画検討の際の参考資料(リファレンス)として一連の技術アセットを提供します。技術アセットには、プロジェクトが採用した技術的アプローチ及び実装方法を整理したドキュメントやAPI仕様、データモデル仕様、オープンソースソフトウェア等が含まれます。
 - また、技術検証レポートでは、技術的知見のみならず、開発技術等を用いて行った技術実証の成果についても共有します。技術実証により得られた当該技術の有効性、制約条件、技術的課題、改善余地、今後の開発への示唆等について知見としてまとめています。
- Project LINKS及びCOMmmONSでは、これらの技術アセットの開発・公開を通じ、地域交通の連携・協働の技術的基盤を提供し、「交通空白」解消など地域交通のR・デザイン全面展開を推進していきます。

Project LINKSとは



- 「Project LINKS」は、データに基づく政策立案の推進(EBPM)や、新たなビジネス創出(オープン・イノベーション)を実現するための、国土交通省の分野横断的なDX推進プロジェクトです。
- これまで活用されてこなかった様々な行政情報を「データ」として再構築し、これを活用できるようにすることで、データに基づく政策立案の推進(EBPM)や、新たなビジネス創出(オープン・イノベーション)の実現を目指しています。
- 行政手続や調査統計など、行政が保有する様々な「情報」をシステムで利用可能な「データ」として再構築するとともに、作成したデータを政策立案や評価、検証等に利用可能とするため、様々なデータ分析ツールを開発します。

地域交通DX推進プロジェクト「COMmmONS」とは



- 「COMmmONS(コモンズ)」は、事業者や地域ごとに業務やシステムなどが独自に構築され、それぞれのサービスやデータが連携していない地域交通の「サイロ化」の課題を解決し、連携・協働を軸とした地域交通のDXを体系的に推進するためのプロジェクトです。
- 具体的には、サービス、データ、マネジメント、ビジネスプロセスの4つの柱で協調領域における相互運用性確保のためのデジタル活用のベストプラクティス創出と、その成果の標準化を一体的に推進することを目的としています。
- コモンズの標準仕様や技術仕様を社会の共通財産として公開・普及させることにより、地域交通の連携・協働の技術的基盤の提供を推進します。

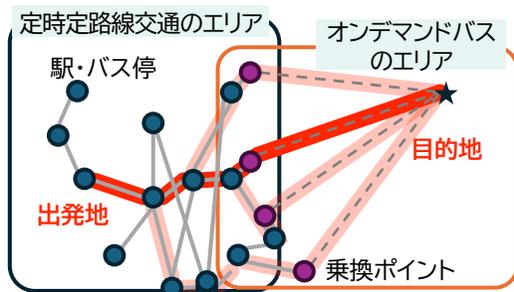
背景・目的

- ・ 深刻な運転手不足に直面する地域交通においては、**定時定路線バスの減便・廃止が進行している**が、「地域の足」を維持していくための方策の一つとして、**オンデマンドバスを導入するケースが増えている**。
- ・ 他方、時刻表などを持たない**デマンド型交通の情報を定時定路線型交通を前提とした既存の経路検索サービスで利用することは難しく**、利用者のアクセシビリティを向上させるうえでの課題となっていた。
- ・ 本プロジェクトでは、経路検索サービスにおいて**デマンド型交通を取り扱うための最新の国際標準データ仕様の有用性を検証する**ため、経路検索アプリケーション及びオンデマンドバス管理システムの開発を行う。
- ・ 具体的には、国際標準化された**GTFS-Flex**及びその発展として研究されている**GTFS-Ondemand**を活用し、一般的な乗換検索アプリをオンデマンドバスに対応させることで、**マルチモーダルな乗換経路検索を実現する**。
- ・ 上記により、オンデマンドバスの利便性向上を図り、持続可能な事業運営の実現を目指す。

開発したシステムの概要

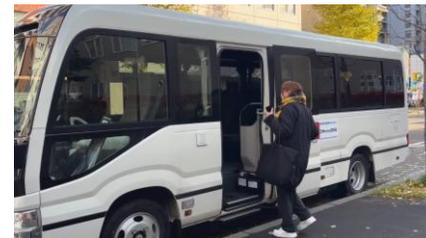
- ・ **GTFS-Flex及びGTFS-Ondemandで定義される全項目を活用、定時定路線交通とオンデマンドバスとにまたがった乗換検索を実現するシステムを開発した**。
- ・ コアとなる乗換検索アルゴリズムは、定時定路線交通の駅・バス停をノードとしたグラフ上にオンデマンドバスとの乗換スポットを重畳させ、グラフ探索アルゴリズムを用いた定時定路線交通の乗換検索と乗換スポットからオンデマンドバスの乗降スポットまでのリアルタイムな所要時間推定を組合せることで実現した。

経路検索アプリで路線バスとオンデマンドバスをまたいだ乗換経路を探す



実証実験の概要

- ・ GTFS-Flex及びGTFS-Ondemandを活用したマルチモーダル乗換検索のベストプラクティスを創出し、「地域の足」「観光の足」としてオンデマンドバスの利便性を向上させるため、**様々な定時定路線交通と組み合わせた実証シナリオを用意し**、北海道札幌市創成イーストエリアで**オンデマンドバスを実際に走行させて実証実験をおこなった**。
- ① Checkpoint型: 時刻表とルートが定まっておらず事前に定められた停車位置のどこからでも乗降するパターン
 - ② Zone-to-point型: サービスエリア内のどこからでも乗車でき事前に定められた停車位置のどこかで降車するパターン
 - ③ Point-to-zone型: 事前に定められた停車位置のどこかで乗車しサービスエリア内で降車するパターン
- ・ Checkpoint型の実証実験は、札幌市民から選んだ**モニター22名を被験者**として、マイクロバス1台とワンボックス型乗合車両2台の合計3台を走行させ、定時定路線交通とオンデマンドバスの乗換えを実際に体験する形式で実施した。



札幌でのフィールド実証の様子



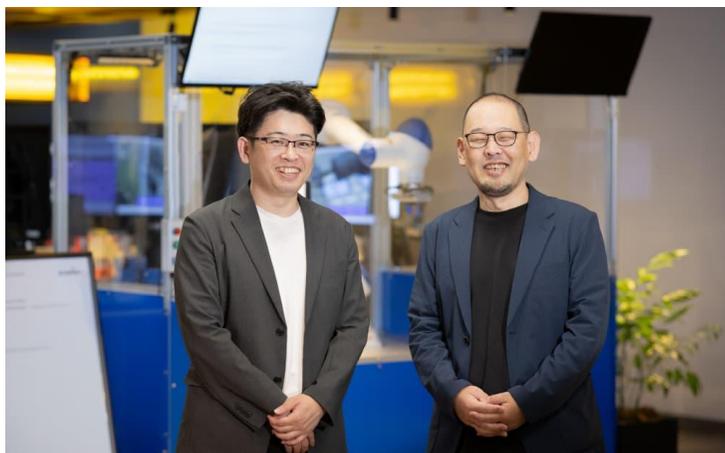
ドライバーアプリ



オンデマンド管理センターの様子

得られた成果

- ・ **GTFS-Flex及びOndemandで定義されたデータ項目を活用**することで、マルチモーダルな乗換検索を実現できるシステムを構築した。
- ・ 実証実験を通じて、GTFS-Flex及びOndemandの活用することで、**一般的な乗換検索アプリ上でオンデマンドバスを含む経路検索を提供**することが、オンデマンドバスの利便性向上に繋がることを示した。



株式会社駅探 RMP事業部長 山田雄太(左)、
TIS株式会社 ソーシャルイノベーション事業部 エキスパート 長井大典(右)

オンデマンド交通を“誰でも使える”ものに

「バスはあるけれど、タイミングが合わない」「駅まで近いのに、ちょっとした坂道がづらい」——。本プロジェクトは、そんな日常の“あと一步”を埋める新しい交通の形を探り、次の時代の“乗車体験”を形づくる取組です。

新しい交通体験を「当たり前」へ

本プロジェクトでは、定時定路線交通も含むマルチモーダルな乗換案内からオンデマンドバスのリアルタイムな配車予約まで、国際標準仕様であるGTFS-FlexやGTFS-Ondemandを活用してシームレスにつながる仕組みを構築し、実証実験を行いました。この実証実験の成果を活用し、特定のアプリに依存せずより自然なUXで、どの経路でも一貫した移動体験の提供を目指しています。

国際標準へのフィードバックで他地域へ展開

乗換案内とオンデマンドバスを連携させた実例として、得られた知見をGTFSの国際標準仕様にフィードバックしていくことで、国際標準仕様への貢献も果たします。

つくって終わりにしない。地域になじむ交通システムに

本プロジェクトは、地域交通を支える事業者の負担を減らし、利用者にとっても使いやすい仕組みを作ることが目的です。この成果が、将来的に「地域交通の使いやすさ」の新しいスタンダードにつながることを期待しています。

本編	
技術検証レポートについて	2
プロジェクトサマリー	3
目次	4
第1章 概要	
解決すべき社会課題と解決アプローチ	6
既存業務フローの課題と目指す業務フロー	7
実現したい価値、想定事業機会	8
本実証実験の全体フロー	9
実施体制・協力事業者一覧	10
第2章 標準仕様調査の方法・結果	
調査の全体像	12
調査結果	13
第3章 開発システム	
システム概要	34
業務フロー	35
システムアーキテクチャ	36
利用技術スタック	37
UI/UX	38
第4章 実証実験	
検証仮説	43
実証実験の全体像	44
KPI	45
実証エリア	46
実証実験の様子	47
実証実験の結果	48
第5章 まとめ	
成果と課題	70
将来展望	72
参考情報・用語集	73
付録	
GTFS-Flex/Ondemand活用システム システム設計書	
GTFS-Ondemand仕様案(日本語訳)	

第1章 概要

運転手不足に直面する地域交通において、代替交通の手段としてオンデマンドバスの導入が広がっている。本プロジェクトでは、国際標準であるGTFS-Flex及びGTFS-Ondemandを活用し、一般的な乗換検索アプリ上でのマルチモーダルな乗換経路検索とリアルタイムの配車予約を実現することで、オンデマンドバスの利用機会増大を図り、持続的な事業運営を可能とすることを目指す。

解決すべき社会課題と解決アプローチ

路線バス減便・廃止による地域の足の担い手不足の対策として、GTFS-Flex及びOndemandを活用し、多様なユーザーのオンデマンドバス利用機会の増大を図る

解決すべき社会課題

国内の公共交通の重要な役割を担う路線バスにおける深刻な運転手不足

- ・ 少子高齢化の進展に加え、いわゆる2024年問題やコロナ禍の影響などにより、定時定路線の路線バスの減便・廃止が進行している。

オンデマンドバスは有効な施策だが独自のUXにより利便性に課題

- ・ 公共ライドシェアや日本版ライドシェア以外にも、オンデマンドバスや乗合タクシーなど、タクシーと路線バスの中のような性質の公共交通も、ユースケースによっては有効だと考えられている。
- ・ オンデマンドバスの実証実験や実導入は全国さまざまな地域で進められており、地域住民には好意的に受け入れられているといわれている。一方で、旅行者なども含め広く一般に認知・利用されるためには、専用アプリやウェブフォームからの検索・予約だけでは限界がある。一般に利用されている経路検索サービス等から、路線バス等の他の交通サービスと統合した形で経路検索や予約ができるUXを実現する必要がある。

解決アプローチ

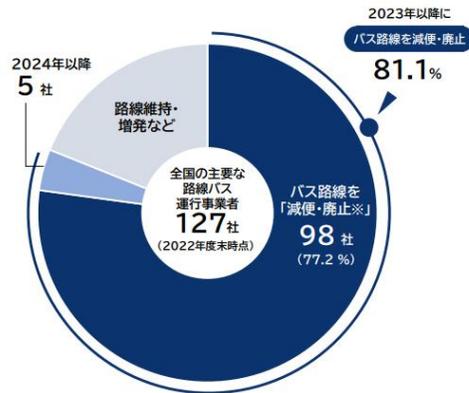
オンデマンドバスの運行を経路検索サービスで表現するためのデータの国際標準仕様であるGTFS-Flex及び国際標準コミュニティで提案されているGTFS-Ondemandの二つの規格を活用し、オンデマンドバスに対応した経路検索サービスを開発することで、以下を実現する。

最適な経路検索の実現

- ・ 国際標準化されたデータ規格を用いてオンデマンド型交通に対応した最適な経路をリアルタイムに検索できるようにすることで、広く一般に普及している経路検索サービスから定時定路線交通とオンデマンドバスを統合した最適な経路案内を可能にし、オンデマンド型交通へのアクセシビリティを向上させる。

経路検索からシームレスな配車予約

- ・ 経路検索からシームレスに配車予約可能なUXを実現することで、オンデマンド型交通へのアクセシビリティを向上させる。



路線バス運行127社の「減便・廃止」動向¹

路線バス 減便・廃止が相次ぐ要因	
2024年問題への対応	<p>現行のダイヤでは、残業規制に対応可能な人繰りが不可能</p> <p>運転手の確保難・既存ドライバーの高齢化 観光・貸切バスへの運転手流出</p>
収益環境の悪化	<p>沿線住民の利用が減少 コロナ禍からの減収分が戻らず、経営を圧迫</p> <p>高速バス・貸切バス事業を犠牲にした路線バス維持策が限界</p>



乗換検索アプリでオンデマンドバスを含む経路検索とオンデマンドバスの予約を実現

経路検索結果から配車予約可能

オンデマンドバスを含む経路検索の実現イメージ

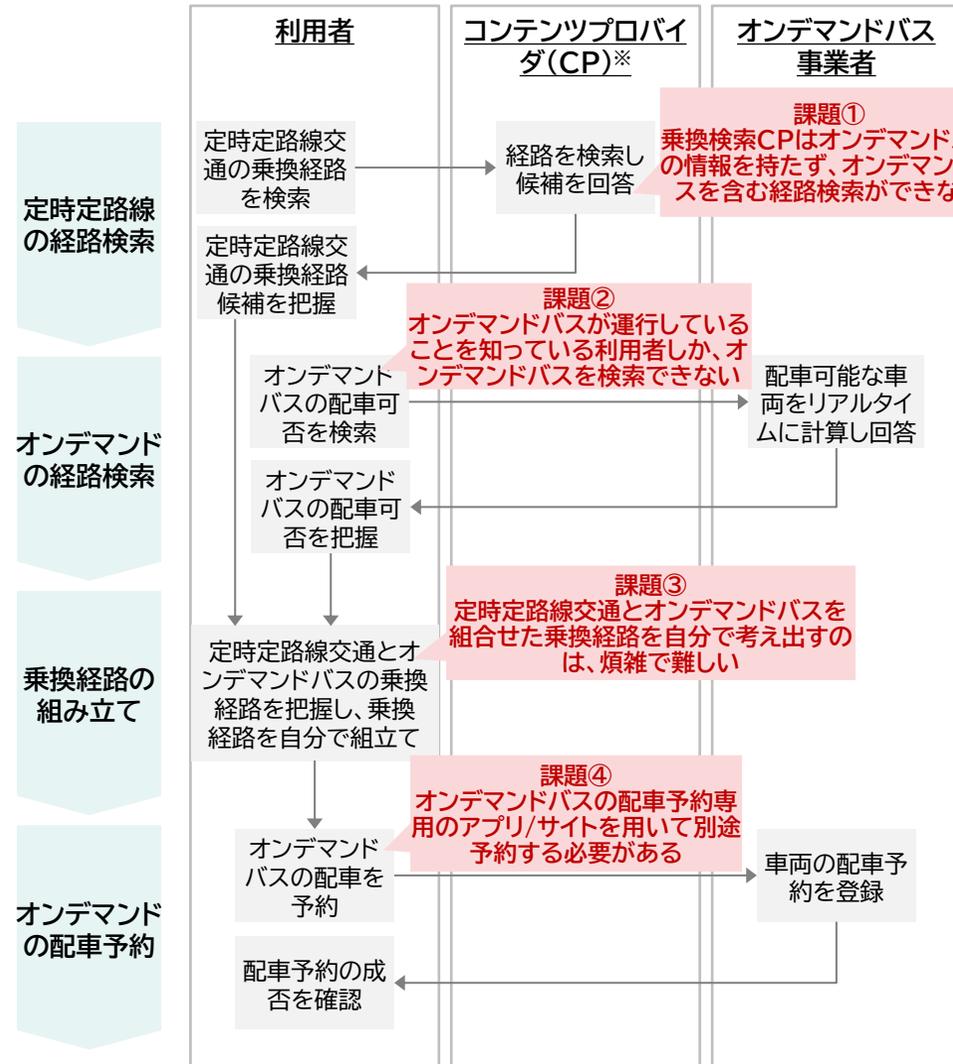
1.帝国データバンク，“特別企画:全国「主要路線バス」運行状況調査(2023年)” |

<https://www.tdb.co.jp/resource/files/assets/d4b8e8ee91d1489c9a2abd23a4bb5219/4991a2d38f614570b04581d9fbdf6690/p231109.pdf>

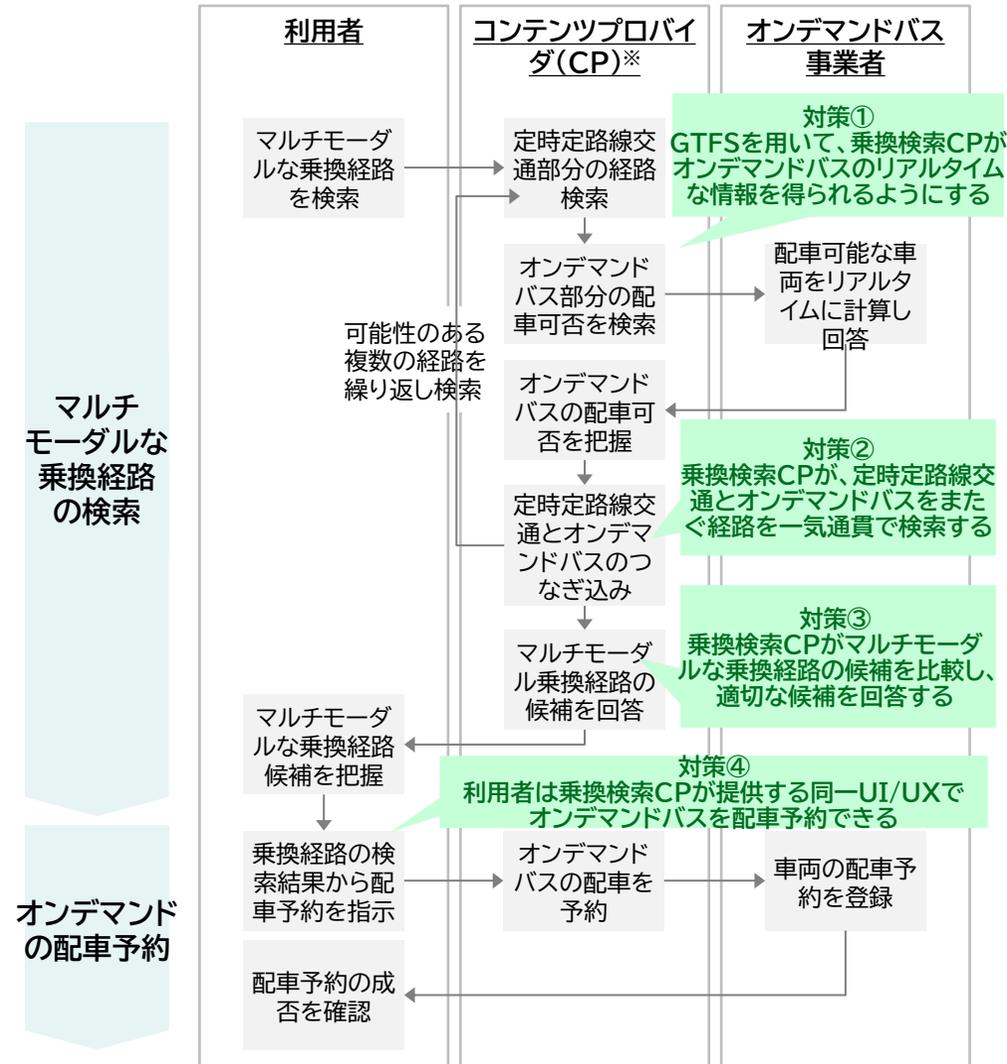
既存業務フローの課題と目指す業務フロー

定時定路線交通とオンデマンドバスにまたがる最適な乗換経路が検索可能かつ、検索結果UIの自然な延長上でオンデマンドバスを予約可能なフローを実現する

既存の業務フロー



目指す業務フロー



※コンテンツプロバイダ(CP):乗換経路の検索サービスを提供する事業者



マルチモーダル乗換検索とオンデマンド配車予約を同一アプリ内で実現することで、オンデマンドバス利用機会を増やし、持続可能な地域の足として事業展開を実現する

実現したい価値

GTFSを活用したシステム間連携によるシンプルな乗換検索と配車予約

- GTFSを活用して乗換検索アプリとオンデマンド配車予約システムを連携させることで、複数のアプリを使いこなして乗換経路を自分で考える必要がなく、使い慣れたアプリのみで定時定路線交通とオンデマンドバスとにまたがった最適な乗換経路検索とオンデマンドバスの配車予約ができるようになる。

特定のアプリに依存しない、オンデマンドバスのリアルタイム経路検索の実現

- オンデマンドバスのリアルタイム経路検索に必要なシステムを設計するに当たって、国際標準であるGTFSに準拠したデータモデルを採用することで、特定のアプリケーション実装に依存しない汎用的なアーキテクチャを開発し、様々なアプリでの実装が可能となる。

オンデマンドバスの利用機会増大による持続可能な事業展開

- 検索や予約など主要なユーザー接点における体験を改善することで、多様なユーザー層によるオンデマンドバスの利用機会を増大させ、収益性の改善など持続可能なオンデマンドバスの運営が可能になる。

1. 使い慣れかつ、2. 特定のアプリに依存しない一般普及したアプリによる乗換検索と、オンデマンドバスの予約



3. 地元住民から観光客まで、多様な層のオンデマンドバス利用機会増による持続可能な事業展開

実現したい価値のイメージ

想定事業機会

利用者

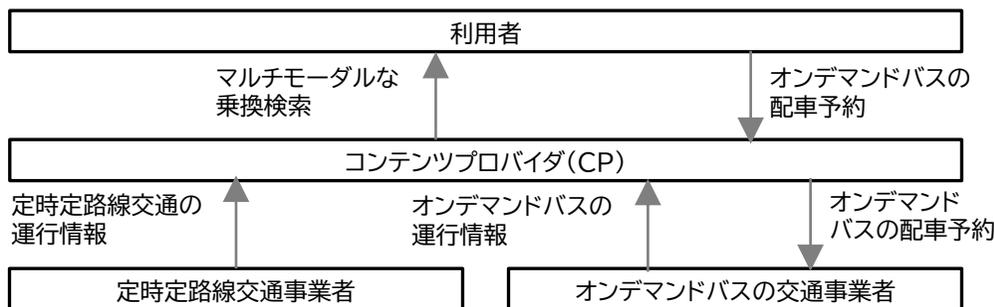
- オンデマンドバスサービスが提供される地域住民
- オンデマンドバスサービスが展開される地域を訪れる旅行者
- 地域の移動の足としてオンデマンドバスの導入を図る地方公共団体
- オンデマンドバスサービスを営む交通事業者

提供価値

- GTFSを用いて乗換検索アプリとオンデマンド配車予約システムを連携させることで、乗換検索アプリからオンデマンドバスを発見、予約できる
- オンデマンドバスの利用機会が増大することにより、オンデマンドバスの収益性が向上する
- オンデマンドバス事業の収益性が改善することにより、地方公共団体の財政負担を軽減できる

サービス展開に向けた仮説

- オンデマンドバスサービスを導入する地方公共団体、交通事業者、その配車予約システムを提供する配車予約サービス事業者及びコンテンツプロバイダと連携し、本成果の社会実装を進める
- GTFS-FlexやOndemandを用いた経路検索アプリ等のインタフェースやデータ項目を公開し、どのような事業者でも再現可能にする



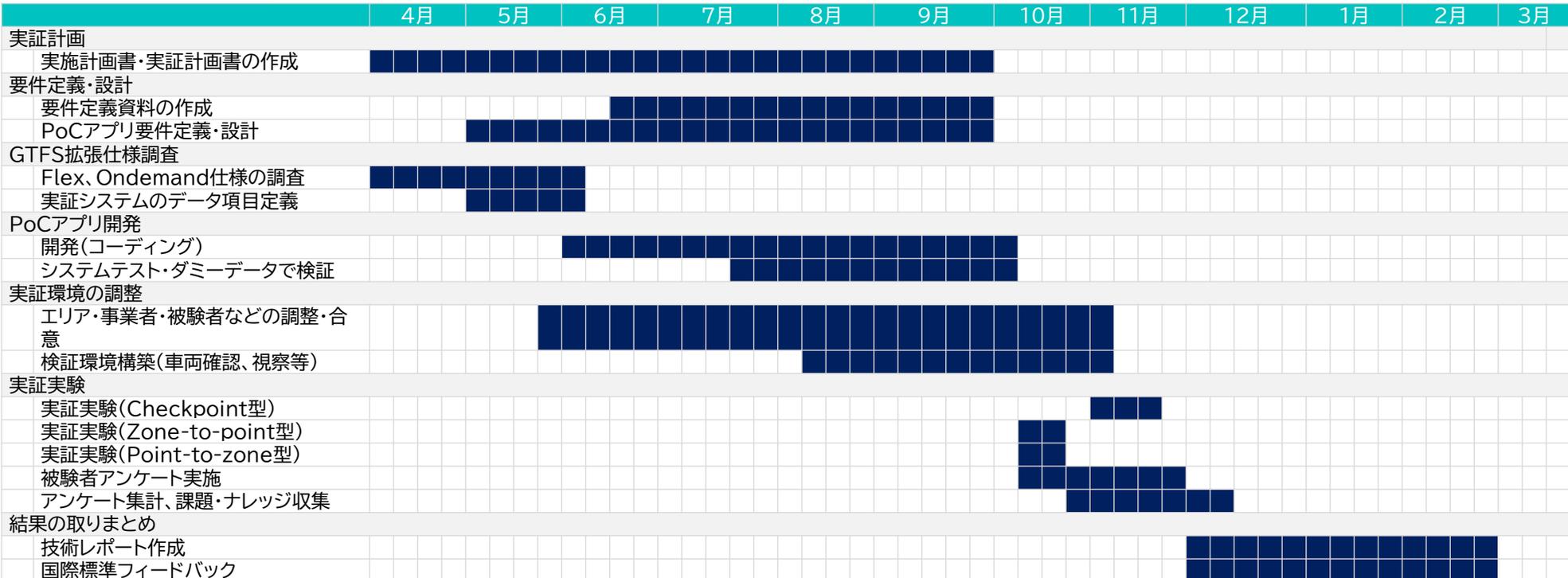
サービスモデル図

オンデマンドバスに対応したGTFS-Flex及びGTFS-Ondemand仕様調査を行い、調査を踏まえたPoCアプリを開発し札幌市でモニターによる実証実験を実施した

本実証実験の業務フロー

実証計画の策定	要件定義・設計	GTFS拡張仕様調査	PoCアプリ開発	実証環境の調整	実証実験	結果の取りまとめ
<ul style="list-style-type: none"> プロジェクトの目的と範囲を定義 関係するGTFS仕様を収集・分析 必要な要件やアルゴリズムを収集・分析 実証実験の検証項目、検証方法、KPI定義 	<ul style="list-style-type: none"> 開発システムの機能要件とアルゴリズム、データインタフェースとユーザーインタフェース、システム構成と非機能要件、実証実験で用いるデータを定義 	<ul style="list-style-type: none"> オンデマンドバスに対応したGTFS-Flex及びGTFS-Ondemand仕様の調査 マルチモーダル乗換検索アルゴリズム及び、実装方法の検討 	<ul style="list-style-type: none"> 要件に合わせ駅探乗換案内の商用サービスを改修 要件に合わせ未来シェアSAVSの商用サービスの追加開発 要件に合わせ必要なGTFSデータを準備 	<ul style="list-style-type: none"> 札幌市と協力して実証計画を合意、市の施策として議会で承認 地域の関係各所と合意形成、車両やドライバーの手配や被験者の募集 	<ul style="list-style-type: none"> 札幌市創成イーストエリアでモニターによる実証実験を実施 アンケート集計、ヒアリングの結果の収集 	<ul style="list-style-type: none"> 技術検証レポート作成 GTFSの国際標準化団体へのフィードバック

本実証実験のスケジュール



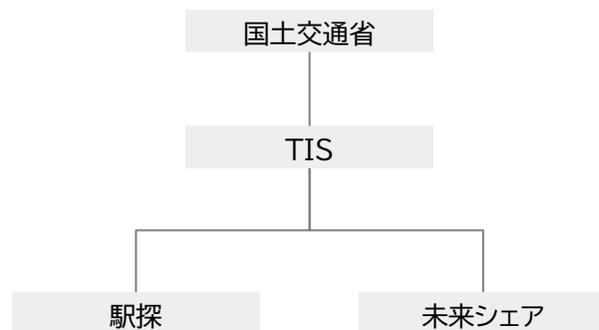
駅探及び未来シェアの商用サービスを改修することで、マルチモーダル乗換検索及びオンデマンド配車予約システムを開発し、札幌市にて実車を用いた実証実験を行った

実施体制

会社名/団体名	担当業務
 国土交通省 総合政策局 情報政策課 / モビリティサービス推進課	<ul style="list-style-type: none"> プロジェクト全体ディレクション
 TIS TIS INTEC Group	<ul style="list-style-type: none"> プロジェクト運営 開発統括/品質管理 実証実験実施支援 ドキュメンテーション
 ekitan	<ul style="list-style-type: none"> 「マルチモーダル乗換検索システム」開発 ※乗換案内サービス「駅探」の改修対応
 MIRAI SHARE	<ul style="list-style-type: none"> 「オンデマンド配車予約システム」開発 ※オンデマンドバス配車予約サービス「SAVS」の改修対応

実証協力事業者

種別	地域	ステークホルダーの名称	役割
自治体	札幌市	 札幌市 City of Sapporo 札幌市まちづくり政策局総合交通計画部	<ul style="list-style-type: none"> 実証フィールドの提供 オンデマンドバスの調達 バス運転手の調達 11月実証のモニター選抜



第2章 標準仕様調査の方法・結果

GTFS-Flex及びGTFS-Ondemandの仕様を調査し、オンデマンドバスの運行要件を定義するために必要となる全ての静的なデータ項目を洗い出した。また、GTFS-Ondemandが拡張したリアルタイムメッセージを用いて、オンデマンドバスのリアルタイムな待ち時間を表現する方法を見いだした。これらの成果と定時定路線交通のグラフ探索アルゴリズムを組合せ、定時定路線交通とオンデマンドバスとにまたがったマルチモーダルな乗換経路を現実的な応答時間で検索可能なアルゴリズムを開発した。

GTFS-Flex及びOndemandのデータ項目やベストプラクティスを調査し、マルチモーダル乗換検索及びオンデマンド配車予約システム実装方法をナレッジ化した

GTFS-Flex及びGTFS-Ondemandとは

- GTFS-Flex: オンデマンド交通を発見しやすくし利用を促進するため、GTFS Scheduleを拡張した仕様。2024年3月に正式採用された。
- GTFS-Ondemand: オンデマンド交通の利用促進のため、GTFS-FlexやGTFS-Fares V2を取り込んだGTFS Schedule及びGTFS Realtimeを発展させた仕様案。

標準データ項目とベストプラクティス調査

#	調査項目名	主要論点	調査手法
1	定時定路線交通との乗換えを考慮した、マルチモーダル乗換検索に関する静的なデータ(GTFS Schedule、GTFS-Flex及びGTFS-Ondemand拡張)の表現技法	GTFS Schedule、GTFS-Flex、及びGTFS-Ondemandを用いた <ul style="list-style-type: none"> 定時定路線交通とオンデマンバスのマルチモーダル乗換検索に関する静的なデータ(サービスエリアや乗降ポイント等)はどのように設計すべきか またその際にデータ項目とその使い方は 	GTFS標準仕様(及びその拡張案)の文献調査 <ul style="list-style-type: none"> GTFS-JP v3 GTFS-Flex GTFS-Ondemand
2	定時定路線交通との乗換えを考慮した、マルチモーダル乗換検索に関する動的なデータ(GTFS-RealtimeのGTFS-Ondemand拡張)の表現技法	<ul style="list-style-type: none"> オンデマンドバスの乗合による遅延を考慮したマルチモーダルな乗換検索アルゴリズムは GTFS-Ondemandリアルタイムメッセージを活用して実装する際のデータ項目とその手法は 	GTFS標準仕様(及びその拡張案)の文献調査 <ul style="list-style-type: none"> GTFS-Ondemand
3	GTFS-Flex及びGTFS-Ondemand拡張を活用したマルチモーダル乗換検索アルゴリズムの検討	<ul style="list-style-type: none"> 定時定路線交通とオンデマンドバスを横断した検索において、利用者にとって違和感のない乗り換えを行うマルチモーダルな乗換経路は GTFS-Flex及びGTFS-Ondemandの静的及び動的なデータを活用して実装するアルゴリズムは 	GTFS標準仕様(及びその拡張案)の文献調査 <ul style="list-style-type: none"> GTFS-Flex GTFS-Ondemand

調査文献

#	文献名	選定理由	URL
1	静的バス情報フォーマット(GTFS-JP)仕様書 [第3版]	GTFS-FlexのベースとなるGTFS Scheduleの国内標準仕様を確認するため	https://www.mlit.go.jp/sogoseisaku/transport/content/001419163.pdf
2	公共交通運行情報データ標準仕様書(GTFS-JP) 第4版(素案)	GTFS-Flexを含むGTFS-JPの最新仕様を確認するため。	https://www.mlit.go.jp/sogoseisaku/transport/sosei_transport_tk_000067.html
3	GTFS Scheduleリファレンス ドキュメント	GTFS Scheduleに取り込まれた最新のGTFS-Flexの仕様を確認するため	https://gtfs.org/ja/documentation/schedule/reference/
4	デマンド レスポンシブ サービス	GTFS-Flexを用いたオンデマンドサービスの定義例を確認するため	https://gtfs.org/ja/documentation/schedule/examples/flex/
5	GTFS-Ondemand	GTFS-Ondemandとして提案されている拡張仕様案を確認するため	https://share.mobilitydata.org/gtfs-ondemand



GTFS-Flex及びOndemandの標準的なデータモデルの活用により、マルチモーダル乗換検索アルゴリズムを実現する方法を導出した

調査結果のまとめ

①GTFS-Flex及びGTFS-Ondemandが拡張した静的なデータ

- GTFS-Flex及びGTFS-Ondemandの仕様を調査し、サービス時間やエリア、乗降スポットや運賃等、オンデマンドバスの運行要件を過不足無く定義し利用者へ適切な情報提供をするために必要となる、以下15ファイルの全ての項目を洗い出した。

ファイル名	概要
1 agency.txt	オンデマンドバスサービスを提供する事業者を定義する。
2 stops.txt	オンデマンドバスの乗降スポットを定義する。
3 routes.txt	オンデマンドバスの路線を定義する。
4 trips.txt	オンデマンドバスの便を定義する。
5 stop_times.txt	location_groupやarealに対して運行時間などを定義する。
6 calendar.txt	運行日を定義する。
7 location_groups.txt	乗降スポットのグループを定義する。
8 location_group_stops.txt	location_groupに含まれる乗降スポットを定義する。
9 location.geojson	任意の地点で乗降できる場合に乗降可能エリアを定義する。
10 booking_rules.txt	オンデマンドバスの予約方法を定義する。
11 wait_rules.txt	オンデマンドバスの待ち時間に関する情報を定義する。
12 booking_deep_links.txt	配車予約アプリなどへのリンク情報を定義する。
13 fare_products.txt	運賃商品の運賃を定義する。
14 fare_leg_rules.txt	運賃商品が適用される区間を定義する。
15 fare_variable_rules.txt	オンデマンドバス特有の変動運賃を定義する。

②GTFS-Ondemandが拡張した動的なデータ

- GTFS-Ondemandが拡張したリアルタイムメッセージ仕様を調査、検索条件に従いオンデマンドバスを配車した場合出発地や到着地で見込まれる待ち時間及び予想運賃をOndemandフィードエンティティを用いて表現する方法を導出した。

③GTFS-Flex及びGTFS-Ondemandによるマルチモーダル乗換検索アルゴリズム

- 以下のアルゴリズムによって、定時定路線交通とオンデマンドバスとにまたがったマルチモーダルな乗換経路が現実的な応答時間で検索可能であると結論付けた。
 - GTFS Scheduleから得たデータや購入した時刻表データを活用し、定時定路線交通の駅・バス停をノードとしたグラフを作成する。
 - そのグラフ上にGTFS-Flex+GTFS-Ondemandの静的なデータによって定義されたオンデマンドバスとの乗換スポットを重畳する。
 - 各乗換スポットに対し、スポットに到達する定時定路線交通の経路をグラフ探索アルゴリズムを用いて算出し、更に乗換スポットからオンデマンドバスの乗降スポットまでの所要時間をGTFS-Ondemandの動的メッセージで得て加算する。
 - 算出された複数の乗換経路を比較し、優先度を付けて利用者へ提示する。

静的データのサンプル

<booking_rules.txt>

```
booking_rule_id,booking_type,message,phone_number,
info_url,booking_url,download_app_url
rule_001,0,駅探アプリから予約をお願いします。080-XXXX-
XXXX,https://maas.ekitan.com/sapporo-
pilot/web/,https://maas.ekitan.com/sapporo-
pilot/download/
```

<wait_rules.txt>

```
wait_rule_id,stop_id,service_id,start_time,end_time,
mean_wait_time,safe_wait_time,max_wait_time
wait_001,lg_001,c_001,09:00:00,16:00:00,15,15,30
wait_002,lg_002,c_001,09:00:00,16:00:00,15,15,30
wait_003,area_001,c_001,09:00:00,16:00:00,15,15,30
```

<location.geojson>

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "id": "area_001",
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [ [
          [141.347451, 43.072256], [141.374002, 43.075841],
          [141.376397, 43.064882], [141.369175, 43.062432],
          [141.363618, 43.056137], [141.361526, 43.056694],
          [141.351734, 43.055288]
        ] ]
      },
      "properties": {
        "stop_name": "創成イーストエリア乗降可能ゾーン",
        "stop_desc": "札幌市創成イーストエリアにおいてオンデマンドバスの乗降が可能なゾーン"
      }
    }
  ]
}
```

動的データのサンプル

<Ondemandリアルタイムメッセージ>

```
{
  "header": {
    "gtfs_realtime_version": "2.0",
    "timestamp": 1748738603
  },
  "entity": {
    "id": "001",
    "on_demand": [
      {
        "wait_time_update": {
          "wait_location":
            "pickup_0001",
          "trip_id": "trip_001",
          "wait_time": 612,
          "max_wait_time": 1512
        }
      },
      {
        "wait_time_update": {
          "wait_location":
            "dropOff_0021",
          "trip_id": "trip_001",
          "wait_time": 1812,
          "max_wait_time": 2712
        }
      },
      {
        "fare_update": {
          "fare_leg_id":
            "fare_leg_001",
          "origin": "pickup_0001",
          "destination":
            "dropOff_0021",
          "amount": 100.0
        }
      }
    ]
  }
}
```

①GTFS-Flex及びGTFS-Ondemandが拡張した静的なデータ

GTFS Scheduleからの拡張内容

- GTFS-Flexは、GTFS Scheduleを拡張しオンデマンド交通の情報を定義可能にした国際標準仕様。オンデマンド交通の情報が標準仕様に従って公開されることで、オンデマンド交通を発見しやすくなり、オンデマンド交通の利用が促進されると期待される。
- GTFS-Ondemandは、GTFS-FlexやGTFS-Fares V2が正式採用されたGTFS Schedule及びGTFS Realtimeを拡張し、待ち時間や変動料金などオンデマンド交通の追加の情報を定義可能にした、国際的に研究が進められている仕様案。
- 「公共交通運行情報標準データ仕様書(GTFS-JP) 第4版」では、GTFS-Flexも正式に取り込まれ仕様として定義された。本プロジェクトでは、GTFS-JP v4に準拠して静的なデータを定義する。

GTFS-Flexが拡張する静的なデータ

	ファイル名	変更/追加	拡張内容
1	stops.txt	変更	GTFS-Flexによって追加されたlocation_groups.txtやlocation.geojsonで定義されるIDと、stops.txtで定義するstop_idは重複せず、一意にならなければならないという制約を追加。
2	routes.txt	変更	デマンド型交通の情報を定義する場合、continuous_pickupやcontinuous_drop_offは利用できないという制約を追加。
3	stop_times.txt	変更	デマンド型交通の情報を定義する場合、arrival timeやdeparture_time等の定時定路線交通用の項目は利用できないという制約を追加。またデマンド型交通のサービス開始時間やサービス終了時間など、デマンド型交通に必要な情報を定義する項目を追加。
4	location_groups.txt	新規	デマンド型交通の乗降スポットのグループを定義。
5	location_group_stops.txt	新規	乗降スポットのグループに属する停留所を定義。
6	location.geojson	新規	デマンド型交通でエリア内の任意の地点で乗降できる場合のエリアを定義。
7	booking_rules.txt	新規	デマンド型交通を予約する方法を定義する。

GTFS-Ondemandが拡張する静的なデータ

	ファイル名	変更/追加	拡張内容
1	agency.txt	変更	android_store_uriやios_store_uriなど、デマンド型交通用のアプリをダウンロードするためのURIが定義する項目を追加。
2	routes.txt	変更	route_typeに13(自動車)を追加する。またデマンド交通の予約アプリを直接開くディープリンクを定義する項目を追加。
3	trips.txt	変更	乗合可否や荷物可否など、デマンド型交通に必要な情報を定義する項目を追加。
4	stop_times.txt	変更	予約を依頼した場所から実際にデマンド型交通を乗降する地点までどの程度近づけるか、また乗降スポットでの平均待ち時間や最大待ち時間など、GTFS-Flexが追加したデマンド型交通の情報をさらに拡張。
5	booking_rules.txt	変更	デマンド型交通の予約アプリのWebサイトやアプリマーケットへのURLを定義する項目を追加。
6	wait_rules.txt	新規	停留所やデマンド型サービスのエリアごとに、平均待ち時間や最大待ち時間などを定義。
7	booking_deep_links.txt	新規	デマンド型交通のアプリを直接開くために使用するAndroidやiOS、Webアプリのディープリンクを定義。
8	fare_leg_rules.txt	変更	Fares v2からの拡張。デマンド型交通の運賃のうち固定料金部分、及び最大料金や最小料金、税込み/税抜き等の情報を定義。
9	fare_variable_rules.txt	新規	デマンド型交通の運賃のうち、変動料金部分を定義。

①GTFS-Flex及びGTFS-Ondemandが拡張した静的なデータ

本実証で利用する静的なデータの項目選定の考え方

- 本プロジェクトで利用する静的なデータは、静的バス情報フォーマット (GTFS-JP)仕様書(第3版)にて必須と定義されているデータ項目に加え、「公共交通運行情報標準データ仕様書(GTFS-JP) 第4版」にて追加されたGTFS-Flexの範囲のデータ項目も網羅するようにデータ項目を選定する。
- GTFS-Ondemandについては、GTFS Scheduleをさらに拡張したデータ項目があるため、こちらについても網羅的に利用した。
- 特に運賃に関しては、GTFS-OndemandがGTFS-Fares V2を拡張しているため、GTFS-Fares V2に基づくfare_products.txt、fare_leg_rules.txtを利用し、GTFS-Ondemandが定めた変動する運賃を定義するために必要な項目を全て網羅するようにデータ項目を選定する。
- また、提供者情報(feed_info.txt)と翻訳情報(translations.txt)については、実証用に閉じた環境で運用し広く公開する予定が無いいため、本プロジェクトでは割愛する。
- 各データ項目へ実際に定義する値については、本プロジェクトの実証要件に基づき、全てのデータ項目を網羅するように定める。

本実証で利用する動的なデータの項目選定の考え方

- 本プロジェクトで利用する動的なデータは、GTFS-OndemandがGTFS Realtimeを拡張したデータ項目を網羅的に利用するように選定する。

本実証における検証事項

- 実証はリアルタイム予約のみで行うが、当日事前予約や前日まで予約の場合はbooking_rules.txtに定義すべき項目が異なる。そのため予約タイプが異なる3パターンのbooking_rules.txtを準備し、当日事前予約や前日まで予約のパターンについては、予約ルールに従って利用者へ提示する予約情報と予約ボタンが押せるタイミングが変わることを確認する。

本実証で利用するファイルとその拡張元、拡張した規格

	ファイル名	拡張元	拡張した仕様
1	agency.txt	GTFS Schedule	GTFS-Ondemand
2	stops.txt	GTFS Schedule	GTFS-Flex
3	routes.txt	GTFS Schedule	GTFS-Flex GTFS-Ondemand
4	trips.txt	GTFS Schedule	GTFS-Ondemand
5	stop_times.txt	GTFS Schedule	GTFS-Flex GTFS-Ondemand
6	calendar.txt	GTFS Schedule	-
7	location_groups.txt	-	GTFS-Flex
8	location_group_stops.txt	-	GTFS-Flex
9	location.geojson	-	GTFS-Flex
10	booking_rules.txt	-	GTFS-Flex GTFS-Ondemand
11	wait_rules.txt	-	GTFS-Ondemand
12	booking_deep_links.txt	-	GTFS-Ondemand
13	fare_products.txt	Fare v2	-
14	fare_leg_rules.txt	Fare v2	GTFS-Ondemand
15	fare_variable_rules.txt	-	GTFS-Ondemand

データ定義と実装サンプル

<trips.txt>

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 route_id	経路ID	○	-	routesから参照。
2 service_id	運行日ID	○	-	calendarから参照。
3 trip_id	便ID	○	-	便を特定するIDを指定。
4 trip_type	便タイプ	-	乗り合い可否を定義する。 0または空 - 乗り合い / 1 - プライベート	0を指定。
5 luggage_allowed	荷物可否	-	乗客の荷物可否を示す。 0または空 - 情報が利用できない / 1 - 荷物を持って乗車可 / 2 - 荷物を持って乗車不可	1を指定。
6 pets_allowed	ペット可否	-	乗客のペット可否を示す。 0または空 - 情報が利用できない / 1 - ペットを連れて乗車可 / 2 - ペットを連れて乗車不可 3 - ペットを連れて乗車できるが、キャリーバッグに入れる必要がある	2を指定。
7 service_animal_allowed	介助動物可否	-	介助動物の可否を示す。 0または空 - 情報が利用できない / 1 - 介助動物を連れて乗車可 / 2 - 介助動物を連れて乗車不可 / 3 - 解除動物を連れて乗車できるが、キャリーバッグに入れる必要がある	2を指定。
8 priced_itinerary	有料旅程可否	-	有料の旅程(高速料金やフェリー料金等)の可否を定義する。 0または空 - 情報が利用できない / 1 - 有料の旅程を使用する場合がある / 2 - 有料の旅程は使用しない	2を指定。

```
route_id,service_id,trip_id,trip_type,luggage_allowed,pets_allowed,service_animal_allowed,priced_itinerary
r_001,c_001,r001_c001_001,0,1,2,2,2
r_001,c_001,r001_c001_011,0,1,2,2,2
r_001,c_001,r001_c001_002,0,1,2,2,2
```

データ定義と実装サンプル

<stop_times.txt> (1/5)

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 trip_id	便ID	○	-	tripsから参照。
2 stop_id	停留所・標柱ID	条件付き ○	location_group_idまたはlocation_idが定義されている場合は禁止。	使用しない。
3 location_id	ロケーションID	条件付き ○	乗客が乗車または降車をリクエストできるゾーンを参照するGeoJSONを示すID。	Zone型の乗降スポットの乗降定義をまとめたレコードに対し、対応するlocations.geojsonのIDを設定。
4 location_group_id	停留所グループID	条件付き ○	乗客が乗車または降車をリクエストできる停留所のグループを示すID。	Checkpoint型の乗降スポットの乗降定義をまとめたレコードに対し、対応するlocation_groupのIDを設定
5 arrival_time	到着時刻	条件付き ○	start_pickup_drop_off_windowまたはend_pickup_drop_off_windowが定義されている場合は禁止。	使用しない。
6 departure_time	出発時刻	条件付き ○	start_pickup_drop_off_windowまたはend_pickup_drop_off_windowが定義されている場合は禁止。	使用しない。
7 stop_sequence	通過順位	○	同一のlocation_groupまたはGeoJSON内で旅行をする際は、stop_times.txtに同じlocation_group_idまたはlocation_idを持つ2つのレコードが必要。	便IDと通過順位を組み合わせて主キーとなるように設定。
8 start_pickup_drop_off_window	オンデマンドサービス開始時間	条件付き ○	デマンド型交通が利用可能になる時間。 location_group_idまたはlocation_idが定義されている場合は必須 end_pickup_drop_off_windowが定義されている場合は必須 arrival_timeまたはdeparture_timeが定義されている場合は禁止	実証開始時間の09:00:00を指定



データ定義と実装サンプル

<stop_times.txt> (4/5)

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
17 drop_off_proximity_level	降車近接レベル	-	リクエスト場所から降車場所までの近接レベルを定義する。 0 または空 - 情報が利用できない / 1 - リクエスト場所にできるだけ近い場所で降車 2 - リクエスト場所から少しの距離の場所(近い交差点等)で降車する場合がある stop_idがGeoJSONを参照する場合のみ利用可能。	stop_idがGeoJSONを参照する場合は1を設定。
18 mean_wait_time	平均待ち時間	-	stop_timeで乗客が待つ平均待ち時間(分)。 stop_idがGeoJSONあるいはlocation_groupを参照し、pickup_type=2あるいは4の場合のみ利用可能。	wait_rule_idフィールドにてwait_rules.txtに定義されているwait_rule_idを指定しているため、mean_wait_timeはwait_ruleに設定された値が利用される。
19 safe_wait_time	安全待ち時間	-	stop_timeで乗客が待つ安全(過去のデータに基づいた95パーセンタイル値)な待ち時間(分)。 stop_idがGeoJSONあるいはlocation_groupを参照し、pickup_type=2あるいは4の場合のみ利用可能。	wait_rule_idフィールドにてwait_rules.txtに定義されているwait_rule_idを指定しているため、safe_wait_timeはwait_ruleに設定された値が利用される。

データ定義と実装サンプル

<stop_times.txt> (5/5)

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
20 max_wait_time	最大待ち時間	-	SLAで定められた最大待ち時間(分)。 stop_idがGeoJSONあるいはlocation_groupを参照し、pickup_type=2あるいは4の場合のみ利用可能。	wait_rule_idフィールドにてwait_rules.txtに定義されているwait_rule_idを指定しているため、max_wait_timeはwait_ruleに設定された値が利用される。
21 wait_rule_id	待機ルールID	-	wait_rule_idが定義されている場合、そのwait_rulesの値でmean_wait_time、safe_wait_time、max_wait_timeは上書きされる。 stop_idがGeoJSONあるいはlocation_groupを参照し、pickup_type=2あるいは4の場合のみ利用可能。	wait_ruleのIDを指定。

```
trip_id,location_id,location_group_id,stop_sequence,start_pickup_drop_off_window,end_pickup_drop_off_window,pickup_type,drop_off_type,pickup_booking_rule_id,drop_off_booking_rule_id,pickup_proximity_level,drop_off_proximity_level,continuous_pickup,continuous_drop_off,mean_wait_time,safe_wait_time,max_wait_time,wait_rule_id
r001_c001_001,,lg_002,1,09:00:00,16:00:00,2,1,rule_001,rule_001,1,1,1,1,15,15,30,wait_002
r001_c001_001,,lg_001,2,09:00:00,16:00:00,1,2,rule_001,rule_001,1,1,1,1,15,15,30,wait_001
r001_c001_011,,lg_001,1,09:00:00,16:00:00,2,1,rule_001,rule_001,1,1,1,1,15,15,30,wait_001
```



データ定義と実装サンプル

<calendar.txt>

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 service_id	運行日ID	○	-	実証期間の運行区分を特定するIDを指定。
2 monday	月曜日	○	-	1(月曜日に運行)を指定。
3 tuesday	火曜日	○	-	1(火曜日に運行)を指定。
4 wednesday	水曜日	○	-	1(水曜日に運行)を指定。
5 thursday	木曜日	○	-	1(木曜日に運行)を指定。
6 friday	金曜日	○	-	1(金曜日に運行)を指定。
7 saturday	土曜日	○	-	1(土曜日に運行)を指定。
8 sunday	日曜日	○	-	1(日曜日に運行)を指定。
9 start_date	サービス開始日	○	-	実証開始日を指定。
10 end_date	サービス終了日	○	-	実証終了日を指定。

```
service_id,monday,tuesday,wednesday,thursday,friday,saturday,sunday,start_date,end_date
c_001,1,1,1,1,1,1,1,1,20251001,20251130
```

<location_groups.txt>

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 location_group_id	停留所グループID	○	location_groupを識別するID。	Checkpoint型の乗降スポットをまとめるために使用。
2 location_group_name	停留所グループ名	-	location_groupの名前。	Checkpoint型の乗降スポットをまとめるために使用。

```
location_group_id,location_group_name
lg_001,乗り換え用location_group
lg_002,サービス用location_group
```

<location_group_stops.txt>

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 location_group_id	停留所グループID	○	location_groupを識別するID。	Checkpoint型の乗降スポットをまとめるために使用。
2 stop_id	停留所・標柱ID	○	そのlocation_groupに所属する停留所・標柱のID。	Checkpoint型の乗降スポットをまとめるために使用。

```
location_group_id,stop_id
lg_001,pd_001
lg_001,pd_002
lg_001,pd_003
```

<locations.geojson> (1/2)

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 Type	タイプ	○	locations.geojsonでは"FeatureCollection"で固定。	サービスエリアの地理的領域を示すために使用。

データ定義と実装サンプル

<locations.geojson> (2/2)

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
2 features[*].id	クエーションID	○	クエーションを示すID。stops.stop_id、locations.geojson id及びlocation_groups.location_group_idの全ての値で一意。	サービスエリアの地理的領域を示すために使用。
3 features[*].type	タイプ	○	locations.geojsonでは"Feature"で固定。	サービスエリアの地理的領域を示すために使用。
4 features[*].properties	属性	○	エリアの属性を指定するオブジェクト。	サービスエリアの地理的領域を示すために使用。
5 features[*].properties.stop_name	クエーション名	-	クエーションを示す名前。	サービスエリアの地理的領域を示すために使用。
6 features[*].properties.stop_desc	クエーション説明	-	クエーションの説明。	サービスエリアの地理的領域を示すために使用。
7 features[*].geometry	形状	○	エリアの形状を指定するオブジェクト。	サービスエリアの地理的領域を示すために使用。
8 features[*].geometry.type	形状タイプ	○	"Polygon"か"MultiPolygon"。本実証ではサービスエリアに飛び地は無いので"Polygon"を指定。	サービスエリアの地理的領域を示すために使用。
9 features[*].geometry.coordinates	座標配列	○	クエーションを定義する地理座標(緯度と経度)の配列。	サービスエリアの地理的領域を示すために使用。

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "id": "area_001",
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [141.347451,43.072256],[141.374002,43.075841],[141.376397,43.064882],
            [141.369175,43.062432],[141.363618,43.056137],[141.361526,43.056694],
            [141.351734,43.055288]
          ]
        ]
      },
      "properties": {
        "stop_name": "サービスエリア",
        "stop_desc": "令和7年度実証実験のサービスエリア"
      }
    }
  ]
}
```



データ定義と実装サンプル

<booking_rules.txt> (1/2)

	フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1	booking_rule_id	予約ルールID	○	予約ルールを識別するID。	予約ルールを特定するIDを指定。
2	booking_type	予約タイプ	○	どの程度前から予約できるかを指定。0:リアルタイム予約 1:事前通知による当日予約まで 2:前日予約まで	0を指定する場合、1を指定する場合、2を指定する場合で画面の表示項目を使い分ける。
3	prior_notice_duration_min	事前通知最小期間	条件付き ○	条件 booking type=1の場合に最小で何分前までに予約が必要か。	booking type=0あるいは2の場合は使用しない。 booking type=1の場合は60を指定(1時間前までOK)
4	prior_notice_duration_max	事前通知最大期間	条件付き ○	条件 booking type=1の場合に最大で何分前から予約ができるか。	booking type=0あるいは2の場合は使用しない。 booking type=1の場合は480を指定(8時間前からOK)
5	prior_notice_last_day	事前通知最終日	条件付き ○	条件 booking type=2の場合に何日前まで予約ができるか。	booking type=0あるいは1の場合は使用しない。 booking type=2の場合は1を指定(前日までOK)
6	prior_notice_last_time	事前通知最終時間	条件付き ○	条件 booking type=2の場合に、prior_notice_last_dayの何時まで予約ができるか。	booking type=0あるいは1の場合は使用しない。 booking type=2の場合は16:00:00を指定(16時までOK)
7	prior_notice_start_day	事前通知開始日	条件付き ○	何日前から予約ができるか。 booking type=0の場合は禁止 booking type=1で prior_notice_duration_maxが定義されている場合は禁止	booking type=0あるいは1の場合は使用しない。 booking type=2の場合は7を指定(7日前からOK)
8	prior_notice_start_time	事前通知開始時間	条件付き ○	条件 prior_notice_start_dayの何時から予約ができるか。	booking type=0あるいは1の場合は使用しない。 booking type=2の場合は09:00:00を指定(9時からOK)

データ定義と実装サンプル

<booking_rules.txt> (2/2)

	フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
9	prior_notice_service_id	事前通知サービスID	条件付き ○	prior_notice_last_dayまたは prior_notice_start_dayをカウントするサービス日を示す。 booking type=0や1では禁止	booking type=0あるいは1の場合は使用しない。 booking type=2の場合はcalendar.txtの service_idを指定
10	message	メッセージ	-	デマンド型交通を予約するときに、サービスを利用する乗客へ提示するメッセージ。	予約ルールに関する簡易的なメッセージを設定。本プロジェクトでは、「実証実験の為、運賃は表示のみで収受いたしません」というメッセージを定義した。
11	pickup_message	乗車メッセージ	-	デマンド型交通で乗車のみを予約するときに、乗客へ提示するメッセージ。	本プロジェクトでは乗車みの配車予約を行わないため、利用しない。
12	drop_off_message	降車メッセージ	-	デマンド型交通で降車のみを予約するときに、乗客へ提示するメッセージ。	本プロジェクトでは降車みの配車予約を行わないため、利用しない。
13	phone_number	電話番号	-	予約するために電話をかける電話番号。	本プロジェクトでの連絡先を記載。
14	info_url	予約情報URL	-	予約ルールに関する情報を提供するURL。	駅探乗換案内のURLを記載。
15	booking_url	予約アプリURL	-	予約リクエストを行うことができるインターネット予約システムまたはアプリへのURL。	駅探乗換案内のURLを記載。
16	download_app_url	アプリダウンロードURL	-	予約リクエストを行うことができるアプリへのURL (Webサイトやアプリマーケット)。	駅探乗換案内の検証用URLを記載。
booking_rule_id,booking_type,message,phone_number,info_url,booking_url,download_app_url rule_001,0,実証実験の為、運賃は表示のみで収受いたしません,080-XXXX-XXXX,https://maas.ekitan.com/sapporo-pilot/web/,https://maas.ekitan.com/sapporo-pilot/web/,https://maas.ekitan.com/sapporo-pilot/download/					



データ定義と実装サンプル

<wait_rules.txt> (1/2)

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 wait_rule_id	待機ルールID	○	待機ルールを識別するID。	待機ルールを特定するIDを指定。
2 stop_id	停留所・標柱ID	○	待機ルールが適用される stops.stop_id、location.geojesonのid、location_groups.location_group_idを示す。	待機ルールを適用する location_groupや location.geojsonのIDを設定。
3 service_id	運行日ID	-	待機ルールが適用されるカレンダーを示す。	この待機ルールを適用する calendarのIDを設定。
4 start_time	適用開始時間	条件付き ○	(stop_times.start_pickup_dropoff_window、stop_times.end_pickup_dropoff_window) 間隔内でなければならない。待機ルールの適用が開始される時間を示す。	実証開始時間の09:00:00を指定。
5 end_time	適用終了時間	条件付き ○	(stop_times.start_pickup_dropoff_window、stop_times.end_pickup_dropoff_window) 間隔内でなければならない。待機ルールの適用が終了される時間を示す。	実証終了時間の16:00:00を指定。
6 mean_wait_time	平均待ち時間	条件付き ○	乗客が待つ平均待ち時間(分)。このフィールドに値が入力されると、stop_times.mean_wait_timeの値がオーバーライドされる。safe_wait_timeおよびmax_wait_timeが指定されていない場合は必須。	本プロジェクトでは15分を設定。

データ定義と実装サンプル

<wait_rules.txt> (2/2)

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
7 safe_wait_time	安全待ち時間	条件付き ○	乗客が待つ安全(過去のデータに基づいた95パーセントイル値)な待ち時間(分)。このフィールドに値が入力されると、stop_times.safe_wait_timeの値がオーバーライドされる。mean_wait_timeおよびmax_wait_timeが指定されていない場合は必須。	本プロジェクトでは15分を設定。
8 max_wait_time	最大待ち時間	条件付き ○	サービスの利用規約で定められた乗客が待つ最大待ち時間(分)。このフィールドに値が入力されると、stop_times.max_wait_timeの値がオーバーライドされる。mean_wait_timeおよびsafe_wait_timeが指定されていない場合は必須。	本プロジェクトでは30分を設定。

データ定義と実装サンプル

<booking_deep_links.txt>

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 booking_deep_link_id	予約ディープリンクID	○	デマンド型交通のアプリを開くために使用するディープリンクを示すID。	ダミーのURLを記載したIDを設定。
2 android_uri	Android URI	-	AndroidのディープリンクをサポートするためにAndroidのIntentと共にAndroidアプリに渡されるURI。Android App Linksが推奨される。	駅探乗換案内の検証用URLを記載。
3 ios_uri	iOS URI	-	iOSで利用されるディープリンクのURI。iOS Universal Linksが推奨される。	駅探乗換案内の検証用URLを記載。
4 web_url	Web URL	-	ウェブブラウザで利用されるURL。	駅探乗換案内のURLを記載。

```
booking_deep_link_id,android_uri,ios_uri,web_url
deeplink_001,http://android.maas.example.com/.well-known/assetlinks.json,https://ios.maas.example.com/.well-known/apple-app-site-association,https://maas.ekitan.com/sapporo-pilot/web/
```

<fare_products.txt>

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 fare_product_id	運賃商品ID	○	運賃商品またはチケット商品のセットを識別する。	運賃商品を識別するIDを設定。
2 amount	通貨金額	○	運賃商品またはチケット商品の金額。	100を設定。
3 currency	通貨コード	○	運賃商品またはチケット商品の通貨。	JPYを設定。

```
fare_product_id,amount,currency
fp_001,100,JPY
```

データ定義と実装サンプル

<fare_leg_rules.txt> (1/2)

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 fare_product_id	運賃商品ID	-	通貨商品のID。	運賃商品を識別するIDを指定。
2 fare_leg_id	運賃区間ID	-	運賃区間を示すID。	運賃区間を特定するIDを指定。
3 amount	運賃	条件付き ○	運賃区間の運賃を示す。運賃変動ルールがこの区間に割り当てられている場合(つまり、行にvariable_group_idが定義されている場合)、運賃の固定価格(または開始価格)として機能する。	開始料金として100を設定
4 trip_cap_amount	最大料金	-	開始料金と変動料金を合算した乗客が支払う最大料金	空白を設定(最大料金は無し)。
5 trip_min_amount	最低料金	-	開始料金と変動料金を合算した乗客が支払う最低料金	最低料金として100を設定。
6 price_adjustment	運賃調整可否	-	運行状況(例:需要の急増)に対応するために運賃調整が行われる可能性があるかどうかを示す。 0 (または空白) - 情報が無い / 1 - 運賃調整が行われる可能性がある / 2 - 運賃調整は行われない	2を指定。
7 tax_inclusion	税込み/税抜き	-	税金が含まれるかどうかを示す。 0 (または空白) - 情報が無い / 1 - 税金が含まれる / 2 - 税金は含まれない	1を指定。
8 tip_acceptance	チップ	-	チップを受け取るかどうかを示す。 0 (または空白) - 情報が無い / 1 - チップを受け取ることができる / 2 - チップは受け取らない	2を指定。
9 variable_group_id	変動運賃グループID	-	変動運賃グループのID。	fare_variable_rulesのvariable_group_idを指定。

```
fare_product_id,fare_leg_id,amount,trip_cap_amount,trip_min_amount,price_adjustment,tax_inclusion,tip_acceptance,variable_group_id
fp_001,fr_001,100,,100,2,1,2,fvg_001
```



データ定義と実装サンプル

<fare_variable_rules.txt>

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 fare_variable_id	変動料金ID	-	変動料金の設定を識別するID。	変動料金を識別するIDを設定。
2 variable_group_id	変動料金グループID	-	変動料金のグループを識別するID。	変動料金のグループを識別するIDを設定。
3 fare_variable_type	変動料金種別	○	変動料金の基準となる変数項目。 0 - キロメートル(距離精算) / 1 - 分(時間精算) / 2 - アクティブ分(車両が走行している時間で精算) / 3 - アイドル分(車両が停止している時間で精算) / 4 - 乗客数 / 5 - 荷物 / 6 - ペット	0(距離精算)を設定。
4 interval	変動料金間隔	条件付き	変動料金が適用される間隔。intervalが0の場合、1回の課金される。 fare_variable_typeが0, 1, 2, 3の場合は必須。それ以外は禁止。	0.25(0.25kmごとに加算される)を設定。
5 start	課金開始値	-	変動料金の適用が開始される値。このフィールドが空の場合、乗車開始と同時に金額が請求される。	0.5(0.5km以上走行したら加算開始)を設定。
6 end	課金終了値	-	変動料金の適用が終了する値。このフィールドが空の場合、乗車が終了するまで金額が請求される。	空白(降車するまで加算を続ける)を設定。
7 amount	単位当たり運賃	条件付き	単位当たりの変動料金。fare_variable_rules.min_amountとfare_leg_rules.max_amountが空の場合は必須。fare_variable_rules.min_amountとfare_leg_rules.max_amountが定義されている場合は禁止。	10を設定。(0.5km以上走行すると、以降0.25kmごとに10円加算)

データ定義と実装サンプル

<booking_deep_links.txt>

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 booking_deep_link_id	予約ディープリンクID	○	デマンド型交通のアプリを開くために使用するディープリンクを示すID。	ダミーのURLを記載したIDを設定。
2 android_uri	Android URI	-	AndroidのディープリンクをサポートするためにAndroidのIntentと共にAndroidアプリに渡されるURI。Android App Linksが推奨される。	駅探乗換案内の検証用URLを記載。
3 ios_uri	iOS URI	-	iOSで利用されるディープリンクのURI。iOS Universal Linksが推奨される。	駅探乗換案内の検証用URLを記載。
4 web_url	Web URL	-	ウェブブラウザで利用されるURL。	駅探乗換案内のURLを記載。

```

booking_deep_link_id,android_uri,ios_uri,web_url
deeplink_001,http://android.maas.example.com/.well-known/assetlinks.json,https://ios.maas.example.com/.well-known/apple-app-site-association,https://maas.ekitan.com/sapporo-pilot/web/
    
```

<fare_products.txt>

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 fare_product_id	運賃商品ID	○	運賃商品またはチケット商品のセットを識別する。	運賃商品を識別するIDを設定。
2 amount	通貨金額	○	運賃商品またはチケット商品の金額。	100を設定。
3 currency	通貨コード	○	運賃商品またはチケット商品の通貨。	JPYを設定。

```

fare_product_id,amount,currency
fp_001,100,JPY
    
```



データ定義と実装サンプル

<fare_leg_rules.txt>

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 fare_product_id	運賃商品ID	-	通貨商品のID。	運賃商品を識別するIDを指定。
2 fare_leg_id	運賃区間ID	-	運賃区間を示すID。	運賃区間を特定するIDを指定。
3 amount	運賃	条件付き ○	運賃区間の運賃を示す。運賃変動ルールがこの区間に割り当てられている場合(つまり、行にvariable_group_idが定義されている場合)、運賃の固定価格(または開始価格)として機能する。	開始料金として100を設定
4 trip_cap_amount	最大料金	-	開始料金と変動料金を合算した乗客が支払う最大料金	空白を設定(最大料金は無し)。
5 trip_min_amount	最低料金	-	開始料金と変動料金を合算した乗客が支払う最低料金	最低料金として100を設定。
6 price_adjustment	運賃調整可否	-	運行状況(例:需要の急増)に対応するために運賃調整が行われる可能性があるかどうかを示す。 0 (または空白) - 情報が無い / 1 - 運賃調整が行われる可能性がある / 2 - 運賃調整は行われない	2を指定。
7 tax_inclusion	税込み/税抜き	-	税金が含まれるかどうかを示す。 0 (または空白) - 情報が無い / 1 - 税金が含まれる / 2 - 税金は含まれない	1を指定。
8 tip_acceptance	チップ	-	チップを受け取るかどうかを示す。 0 (または空白) - 情報が無い / 1 - チップを受け取ることができる / 2 - チップは受け取らない	2を指定。
9 variable_group_id	変動運賃グループID	-	変動運賃グループのID。	fare_variable_rulesのvariable_group_idを指定。

fare_product_id,fare_leg_id,amount,trip_cap_amount,trip_min_amount,price_adjustment,tax_inclusion,tip_acceptance,variable_group_id
fp_001,fr_001,100,,100,2,1,2,fvg_001

データ定義と実装サンプル

<fare_variable_rules.txt> (1/2)

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 fare_variable_id	変動料金ID	-	変動料金の設定を識別するID。	変動料金を識別するIDを設定。
2 variable_group_id	変動料金グループID	-	変動料金のグループを識別するID。	変動料金のグループを識別するIDを設定。
3 fare_variable_type	変動料金種別	条件付き ○	変動料金の基準となる変数項目。 0 - キロメートル(距離精算) / 1 - 分(時間精算) / 2 - アクティブ分(車両が走行している時間で精算) / 3 - アイドル分(車両が停止している時間で精算) / 4 - 乗客数 / 5 - 荷物 / 6 - バット	0(距離精算)を設定。
4 interval	変動料金間隔	条件付き ○	変動料金が適用される間隔。intervalが0の場合、1回のみ課金される。 fare_variable_typeが0, 1, 2, 3の場合は必須。それ以外は禁止。	0.25(0.25kmごとに加算される)を設定。
5 start	課金開始値	-	変動料金の適用が開始される値。このフィールドが空の場合、乗車開始と同時に金額が請求される。	0.5(0.5km以上走行したら加算開始)を設定。
6 end	課金終了値	-	変動料金の適用が終了する値。このフィールドが空の場合、乗車が終了するまで金額が請求される。	空白(降車するまで加算を続ける)を設定。
7 amount	単位当たり運賃	条件付き ○	単位当たりの変動料金。 fare_variable_rules.min_amountと fare_leg_rules.max_amountが空の場合は必須。 fare_variable_rules.min_amountと fare_leg_rules.max_amountが定義されている場合は禁止。	10を設定。(0.5km以上走行すると、以降0.25kmごとに10円加算)



データ定義と実装サンプル

<fare_variable_rules.txt>

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
8 min_amount	最低推定運賃	条件付き ○	最低推定運賃。正確な金額が不明の場合に使用する。fare_variable_rules.amountが定義されている場合は禁止。fare_variable_rules.max_amountと併用が必須。	amountを定義するため、使用しない。
9 max_amount	最大推定運賃	条件付き ○	最大推定運賃。正確な金額が不明の場合に使用する。fare_variable_rules.amountが定義されている場合は禁止。fare_variable_rules.min_amountと併用が必須。	amountを定義するため、使用しない。
10 currency	通貨	条件付き ○	運賃の通貨を指定する。fare_variable_rules.amount、fare_variable_rules.min_amount、fare_variable_rules.max_amountのいずれかが定義されている場合は必須。それ以外は禁止。	JPYを指定。
11 price_adjustment	料金調整可否	-	運行状況(例:需要の急増)に対応するために運賃調整が行われる可能性があるかどうかを示す。0 (または空白) - 情報が無い / 1 - 運賃調整が行われる可能性がある / 2 - 運賃調整は行われない	2を指定。
12 tax_inclusion	税込み/税抜き	-	税金が含まれるかどうかを示す。0 (または空白) - 情報が無い / 1 - 税金が含まれる / 2 - 税金は含まれない	1を指定。

fare_variable_id,variable_group_id,fare_variable_type,interval,start,end,amount,currency,price_adjustment,tax_inclusion
 fv_001,fvg_001,0,0.25,0.5,,10,JPY,2,1

データ定義と実装サンプル

<agency.txt>

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 agency_id	事業者ID	○	-	事業者(TIS)の法人番号を設定。
2 agency_name	事業者名称	○	-	事業者(TIS)の名称を設定。
3 agency_url	事業者URL	○	-	事業者(TIS)HPのトップページのURLを設定。
4 agency_timezone	タイムゾーン	○	-	「Asia/Tokyo」を設定。
5 agency_lang	言語	-	-	「ja」を設定。
6 agency_phone	電話番号	-	-	事業者(TIS)の窓口電話番号を設定。

agency_id,agency_name,agency_url,agency_timezone,agency_lang,agency_phone
 201001134133,TIS(株),https://www.tis.co.jp/,Asia/Tokyo,ja,03-5337-7070

<stops.txt>

フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1 stop_id	停留所・標柱ID	○	stops.stop_idを通じて一意だけでなく、location.geojesonのid、及びlocation_groups.location_group_idの全てを通じて一意となるように設定する。	Checkpoint型の乗降スポットとして利用。
2 stop_name	停留所・標識名称	○	-	Checkpoint型の乗降スポット名を指定。
3 stop_lat	緯度	○	-	Checkpoint型の乗降スポットの緯度を指定。
4 stop_lon	経度	○	-	Checkpoint型の乗降スポットの経度を指定。

stop_id,stop_name,stop_lat,stop_lon
 s_001,アリオ札幌,43.0731,141.3713
 s_002,サッポロビール園,43.0726,141.3683
 s_003,北新病院,43.0723,141.3609



データ定義と実装サンプル

<routes.txt>

	フィールド名	日本語名	必須	拡張内容の説明	本プロジェクトでの利用状況
1	route_id	経路ID	○	-	一意なIDを設定。
2	agency_id	事業者ID	○	-	agencyのIDを参照。
3	route_short_name	経路略称	条件付き ○	-	route_long_nameを設定するため使用しない。
4	route_long_name	経路名	条件付き ○	-	本実証名を設定。
5	continuous_pickup	連続乗車	条件付き ○	当該routeに属するstop_timesにてデマンド型交通が定義されている場合 合は無し(1または空白)にしなければならない。	1を設定。
6	continuous_drop_off	連続降車	条件付き ○	当該routeに属するstop_timesにてデマンド型交通が定義されている場合 合は無し(1または空白)にしなければならない。	1を設定。
7	route_type	経路タイプ	○	交通事業者の種類として、13(自動車)を追加。	3(バス)を使用。
8	booking_deep_link_id	予約ディープリンクID	-	デマンド型交通のアプリを開くために使用するディープリンクを示すID。	booking_deep_linkのIDを参照。

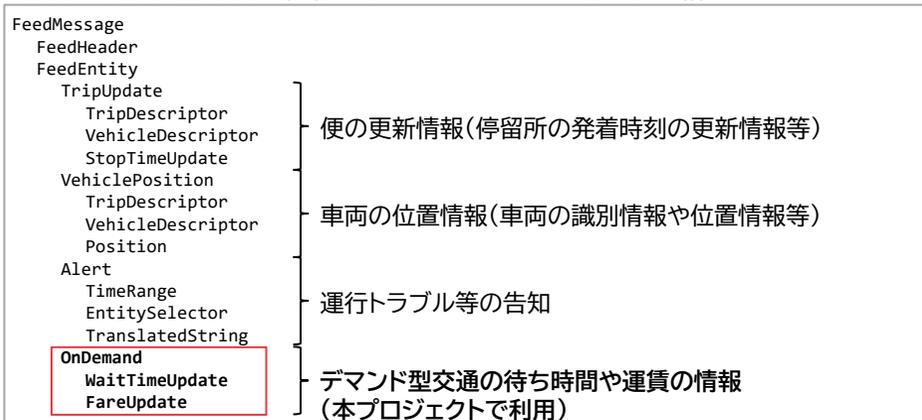
```
route_id,agency_id,route_short_name,route_long_name,
continuous_pickup,continuous_drop_off,route_type,booking_deep_link_id
r_001,2010001134133,,2025年度札幌市実証用オンデマンドバス,1,1,3,deeplink_001
```

②GTFS-Ondemandが拡張した動的なデータ

GTFS Realtimeからの拡張内容

- GTFS-OndemandはGTFS Realtimeを拡張し、デマンド型交通の待ち時間や運賃に関するリアルタイムな情報を提供するOnDemandメッセージを定義する。
- GTFS Realtimeが従来から定義するTripUpdateメッセージやVehiclePositionメッセージ、Alertメッセージとは排他的関係となっており、OnDemandメッセージはそれ単体で提供する必要がある。

GTFS-Ondemandが拡張したリアルタイムメッセージの構造



本実証で利用するデータ項目の選定手法と定義

- 本プロジェクトでは、GTFS-Ondemandが拡張した仕様案に従ってProtocol Buffersの定義ファイルを作成し、オンデマンドバス配車予約システムがリアルタイムに推定する乗車スポット及び降車スポットでの待ち時間、及びその運賃をProtocol Buffersのリアルタイムメッセージとして提供する。
- なお、本プロジェクトでは、乗降スポットにおいて車両が配車されないという状況は発生し得ないため、指定された乗降スポットで利用できる車両の多寡を示すVehicleAvailabilityは利用しない。

OnDemandメッセージのProtocol Buffers定義

```

extend FeedEntity {
  optional OnDemand on_demand = 1000;
}

message OnDemand {
  message WaitTimeUpdate {
    repeated string wait_location = 1;
    optional string trip_id = 2;
    repeated string s2_cell = 3;
    optional int32 wait_time = 4;
    optional int32 safe_wait_time = 5;
    optional int32 max_wait_time = 6;

    enum VehicleAvailability {
      NO_VEHICLES = 1;
      LOW = 2;
      MEDIUM = 3;
      HIGH = 4;
    }
    optional VehicleAvailability vehicle_availability = 7;
  }

  message FareUpdate {
    optional string fare_leg_id = 1;
    repeated string origin = 2;
    repeated string destination = 3;
    repeated string s2_origin = 4;
    repeated string s2_destination = 5;
    required float amount = 6;

    message FareVariableAmount {
      required string fare_variable_id = 1;
      required float amount = 2;
    }
    repeated FareVariableAmount fare_variable_amount = 7;
  }

  repeated WaitTimeUpdate wait_time_update = 1;
  repeated FareUpdate fare_update = 2;
}

```

データ定義と実装サンプル

<FeedMessageメッセージ>

フィールド名	型	必須	多重度	拡張内容の説明	本プロジェクトでの利用状況
1 header	FeedHeader	○	1	このフィードに関するメタデータ。	フィードのバージョンやタイムスタンプを設定。
2 entity	FeedEntity	条件付き ○	複数	フィードの内容。	onDemandメッセージを設定。

<FeedHeaderメッセージ>

フィールド名	型	必須	多重度	拡張内容の説明	本プロジェクトでの利用状況
1 gtfs_realtime_version	string	○	1	フィードのバージョン。	現在のバージョンである2.0を指定。
2 timestamp	string	○	1	フィードが作成された瞬間のタイムスタンプ。	フィードが作成された日時のPOSIX時間を設定。

<FeedEntityメッセージ>

フィールド名	型	必須	多重度	拡張内容の説明	本プロジェクトでの利用状況
1 id	string	○	1	フィード固有の識別子。	重複しない番号を採番して設定。
2 trip_update	TripUpdate	条件付き ○	1	便の出発遅延に関するリアルタイムのデータ。 trip_update、vehicle、alertのいずれか1つ以上のフィールドを指定するか、on_demandのみを指定する。	本プロジェクトでは使用しない。
3 vehicle	VehiclePosition	条件付き ○	1	車両のリアルタイムな位置情報。 trip_update、vehicle、alertのいずれか1つ以上のフィールドを指定するか、on_demandのみを指定する。	本プロジェクトでは使用しない。

データ定義と実装サンプル

<OnDemandメッセージ>

フィールド名	型	必須	多重度	拡張内容の説明	本プロジェクトでの利用状況
4 alert	Alert	条件付き ○	1	何らかのインシデントが発生したことを示すデータ。 trip_update、vehicle、alertのいずれか1つ以上のフィールドを指定するか、on_demandのみを指定する。	本プロジェクトでは使用しない。
5 on_demand	OnDemand	条件付き ○	複数	オンデマンドサービスに関するリアルタイムのデータ。 trip_update、vehicle、alertのいずれか1つ以上のフィールドを指定するか、on_demandのみを指定する。	WaitTimeUpdateメッセージ及びFareUpdateメッセージを設定する。

<OnDemandメッセージ>

フィールド名	型	必須	多重度	拡張内容の説明	本プロジェクトでの利用状況
1 wait_time_update	WaitTimeUpdate	-	複数	オンデマンド待ち時間のリアルタイム更新情報。	出発地及び到着地での待ち時間を示すために使用。
2 fare_update	FareUpdate	-	複数	オンデマンド運賃のリアルタイム更新情報。	オンデマンドバスの運賃を示すために使用。



データ定義と実装サンプル

<WaitTimeUpdateメッセージ> (1/2)

フィールド名	型	必須	多重度	拡張内容の説明	本プロジェクトでの利用状況
1 wait_location	string	条件付き ○	複数	stops.stop_id、location_groups.location_group_id、またはlocations.geojsonのIDを参照。オンデマンドの位置または位置グループのIDs2_cellフィールドが指定されていない場合は必須。	出発地あるいは目的地のIDを設定。
2 trip_id	string	-	1	待ち時間更新に関するtrip_id。	オンデマンドバスのtrip_idを設定。
3 s2_cell	string	条件付き ○	複数	S2CellIDを参照。エリアをカバーする1つ以上のS2セルのID。wait_locationフィールドが指定されていない場合は必須。	本プロジェクトでは使用しない。
4 wait_time	int32	条件付き ○	1	利用者がその場所で待つ待ち時間(秒)。wait_time、safe_wait_time、max_wait_time、vehicle_availabilityのいずれか1つ以上必須。	推定されたその場所での待ち時間を設定。
5 safe_wait_time	int32	条件付き ○	1	利用者がその場所で待つ安全(過去のデータに基づいた95パーセント値)な待ち時間(秒)。wait_time、safe_wait_time、max_wait_time、vehicle_availabilityのいずれか1つ以上必須。	本プロジェクトでは使用しない。

データ定義と実装サンプル

<WaitTimeUpdateメッセージ> (2/2)

フィールド名	型	必須	多重度	拡張内容の説明	本プロジェクトでの利用状況
6 max_wait_time	int32	条件付き ○	1	利用者がその場所で待つ最大の待ち時間(秒)。wait_time、safe_wait_time、max_wait_time、vehicle_availabilityのいずれか1つ以上必須。	推定されたその場所での最大の待ち時間を設定。
7 vehicle_availability	VehicleAvailability	条件付き ○	1	その場所での車両の利用可能性。wait_time、safe_wait_time、max_wait_time、vehicle_availabilityのいずれか1つ以上必須。	本プロジェクトでは使用しない。



データ定義と実装サンプル

<FareUpdateメッセージ>(1/2)

フィールド名	型	必須	多重度	拡張内容の説明	本プロジェクトでの利用状況
1 fare_leg_id	string	-	1	運賃区間ID。fare_leg_rules.fare_leg_idと一致する必要がある。	オンデマンドバスの運賃ルールのはfare_leg_idを設定。
2 origin	string	条件付き ○	複数	stops.stop_id、location_groups.location_group_id、またはlocations.geojsonのIDを参照。出発地の位置または位置グループのID。s2_originフィールドが指定されている場合は禁止。	出発地のIDを設定。
3 destination	string	条件付き ○	複数	stops.stop_id、location_groups.location_group_id、またはlocations.geojsonのIDを参照。目的地の位置または位置グループのID。s2_destinationフィールドが指定されている場合は禁止。	目的地のIDを設定。
4 s2_origin	string	条件付き ○	複数	S2CellIDを参照。出発地をカバーする1つ以上のS2セルID。originフィールドが指定されている場合は禁止。	本プロジェクトでは使用しない。

データ定義と実装サンプル

<FareUpdateメッセージ>(2/2)

フィールド名	型	必須	多重度	拡張内容の説明	本プロジェクトでの利用状況
5 s2_destination	string	条件付き ○	複数	S2CellIDを参照。目的地をカバーする1つ以上のS2セルのID。destinationフィールドが指定されている場合は禁止。	本プロジェクトでは使用しない。
6 amount	非負のfloat	○	1	運賃のリアルタイムな基本料金。	推定された運賃を設定。
7 fare_variable_amount	FareVariableAmount	条件付き ○	複数	運賃変数タイプごとのリアルタイム単価。fare_leg_idが定義されていて、variable_group_idに複数の変数がある場合は必須。	本プロジェクトでは変動運賃として距離精算のみ定義するため、使用しない。

```

{
  "header": {
    "gtfs_realtime_version": "2.0",
    "timestamp": 1748738603
  },
  "entity": {
    "id": "001",
    "on_demand": [{
      "wait_time_update": {
        "wait_location": "pickup_0001",
        "trip_id": "trip_001",
        "wait_time": 612,
        "max_wait_time": 1512
      }
    }],
    "fare_update": {
      "fare_leg_id": "fare_leg_001",
      "origin": "pickup_0001",
      "destination": "dropOff_0021",
      "amount": 100.0
    }
  }
}
    
```

③GTFS-Flex及びGTFS-Ondemandを活用したマルチモーダル乗換検索アルゴリズム

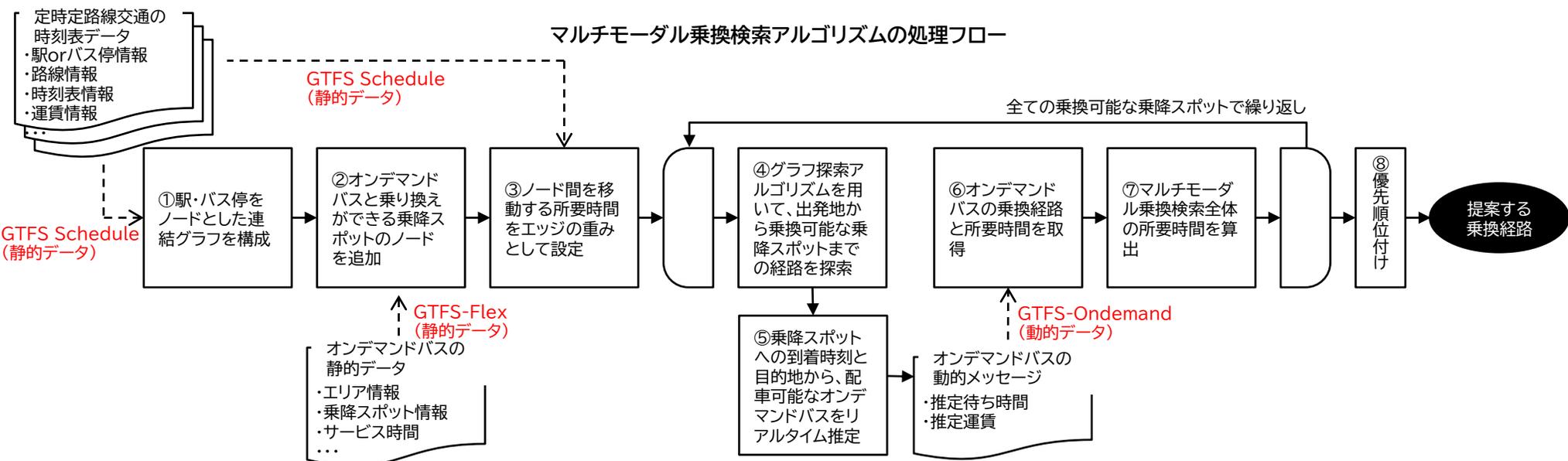
グラフ探索アルゴリズムとリアルタイム配車計算の融合

- ・グラフ探索アルゴリズムを用いて定時定路線交通の乗換経路を検索し、そこにオンデマンドバスのリアルタイム配車計算を以下のような手順でつなぎ込むことで、現実的な検索時間でマルチモーダルな乗換検索が実現できることがわかった。

マルチモーダル乗換検索アルゴリズムの処理概要

- ① 定時定路線交通の乗換経路を検索するために、駅・バス停をノードとし電車やバスの便が設定されている駅・バス停間及び徒歩で乗り換えができる駅・バス停間をエッジとして接続した連結グラフを作成する。
- ② ①で作成した連結グラフへ、GTFS-Flexから得た定時定路線交通とオンデマンドバスの乗り換えができる乗降スポットのノードを追加し、乗換対象となる駅・バス停とエッジで接続する。
定時定路線交通とオンデマンドバス間を乗り換える場合、必ずいずれかの乗換可能な乗降スポットを経由するルールを定めることで、オンデマンドバスの所要時間を推定する回数を乗換可能な乗降スポット数以下に限定することができる。
- ③ 駅・バス停間や駅・バス停から乗降スポットなど、ノード間を移動する移動所要時間をエッジの重みとして設定する。

- ④ 定時定路線交通の探索グラフ上の出発地と各乗換可能な乗降スポット間の乗換経路をDijkstra法等のグラフ探索アルゴリズムを用いて探索する。
- ⑤ 各乗換可能な乗降スポットとその到着時刻、オンデマンドバスの目的地を条件として、乗降スポットから目的地へと至る配車可能なオンデマンドバスをリアルタイムに推定。
- ⑥ オンデマンドバスの乗換経路とその所要時間をGTFS-Ondemandのリアルタイムデータとして取得する。
- ⑦ 乗換可能な乗降スポットごとに、定時定路線交通の所要時間とオンデマンドバスの推定所要時間を加算することで、そのマルチモーダル乗換経路全体の所要時間を算出する。
- ⑧ 所要時間が短い順など、検索されたマルチモーダル乗換経路に優先順位を付け、利用者へ提案する。



④Flexのみマルチモーダル経路探索アルゴリズム

リアルタイム配車計算を用いないマルチモーダル経路探索のアルゴリズム

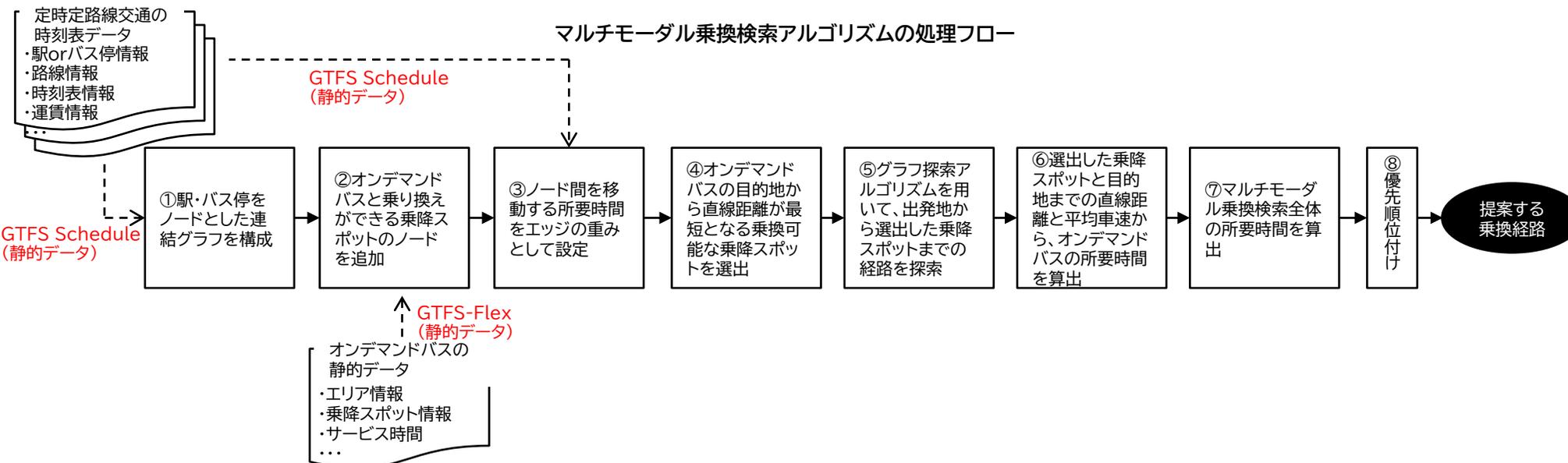
- ・グラフ探索アルゴリズムとリアルタイム配車計算を融合させたマルチモーダル経路探索アルゴリズムと比較するために、GTFS-Flexによるオンデマンドバスの静的な配車定義をグラフ探索アルゴリズムと組み合わせた経路探索も実現した。

マルチモーダル乗換検索アルゴリズムの処理概要

- ① 定時定路線交通の乗換経路を検索するために、駅・バス停をノードとし電車やバスの便が設定されている駅・バス停間及び徒歩で乗り換えができる駅・バス停間をエッジとして接続した連結グラフを作成する。
- ② ①で作成した連結グラフへ、GTFS-Flexから得た定時定路線交通とオンデマンドバスの乗り換えができる乗降スポットのノードを追加し、乗換対象となる駅・バス停とエッジで接続する。
GTFS-Flexのみを用いる場合、オンデマンドバスの配車エリア上の目的地(出発地)と各乗換ポイント間の所要時間をリアルタイムに算出することができないため、目的地(出発地)と直線距離(ユークリッド距離)が最短の乗換ポイントからオンデマンドバスを利用するものとする。
- ③ 駅・バス停間や駅・バス停から乗降スポットなど、ノード間を移動する移動所要時間をエッジの重みとして設定する。

- ④ 定時定路線交通と乗換可能な乗降スポットのうち、オンデマンドバスの目的地と直線距離が最も短くなる乗降スポットを一つ選出する。
- ⑤ 定時定路線交通の探索グラフ上の出発地と選出した乗降スポット間の乗換経路をDijkstra法等のグラフ探索アルゴリズムを用いて探索する。
- ⑥ オンデマンドバスの所要時間は、選出した乗降スポットとオンデマンドバスの目的地間の直線距離を、事前に設定したオンデマンドバスの平均車速で割って算出する。
- ⑦ 定時定路線交通の所要時間と算出したオンデマンドバスの所要時間を加算することで、そのマルチモーダル乗換経路全体の所要時間を算出する。
- ⑧ 所要時間が短い順など、検索されたマルチモーダル乗換経路に優先順位を付け、利用者へ提案する。

マルチモーダル乗換検索アルゴリズムの処理フロー



第3章 開発システム

商用サービス(駅探乗換案内及び未来シェアSAVS)を基に、GTFS-Flex及びGTFS-Ondemandにて定義されたデータ項目を活用して、定時定路線交通とオンデマンドバスとにまたがったマルチモーダルな乗換検索を行い、検索結果から直接リアルタイムにオンデマンドバスの配車予約ができるシステムを開発した。

駅探乗換案内と未来シェアSAVSを基に、国際標準規格であるGTFS-Flex及びOndemandを活用したマルチモーダル乗換検索&配車予約システムを開発した

システム概要

開発スコープ

本プロジェクトでは、広く使われている乗換案内のUI/UXから定時定路線交通とオンデマンドバスとにまたがるマルチモーダルな乗換検索を行い、検索結果からシームレスにリアルタイムでオンデマンドバスの配車予約できるシステムを開発した。

また、GTFS-Flex及びGTFS-Ondemandにて定義されたオンデマンドバスの運行要件に過不足無く対応するために必要な全ての項目について実装・データ連携し、各項目が利用者に理解できるよう駅探乗換案内のUI改修も行った。

オンデマンドバスの運行形態としては、時刻表とルートが定まっておらず事前に定められた停車位置のどこからでも乗降できるCheckpoint型、サービスエリア内のどこからでも乗車でき事前に定められた停車位置のどこかで降車するZone-to-point型、乗車と降車がその逆となるPoint-to-zone型といったユースケースが想定されるが、これらのユースケースに全て対応できるシステムを構築した。

実現方法

本プロジェクトのコアとなる定時定路線交通とオンデマンドバスとにまたがるマルチモーダル乗換検索は、定時定路線交通の駅・バス停をノードとしたグラフ上にオンデマンドバスとの乗換スポットを重畳させ、グラフ探索アルゴリズムを用いた定時定路線交通の乗換検索と乗換スポットからオンデマンドバスの乗降スポットまでのリアルタイムな所要時間推定を組み合わせることで実現した。

具体的には、開発したシステムは“駅探乗換案内”、“SAVS”、“SAVSドライバーアプリ”という3つのコンポーネントで構成され、それらの間はAPIで接続される。“駅探乗換案内”は定時定路線交通とオンデマンドバスとにまたがるマルチモーダル乗換検索を行い、利用者が指示した乗換経路上のオンデマンドバスの配車予約をSAVSへ依頼する。“SAVS”はオンデマンドバスの車両情報や配車予約情報を統括し、オンデマンドバスのリアルタイムな現在位置や今後の行き先、配車予約の条件などから配車計算を行い、オンデマンドバスへ適切な指示を出す。

最後の“SAVSドライバーアプリ”はオンデマンドバスのダッシュボードに据え付けられたタブレットに内蔵され、車両のリアルタイム情報を収集するとともに、次の行き先指示や利用者の乗降記録など、ドライバーとのインタフェースの役割も果たす。

システムイメージ



駅探アプリ

予約内容をSAVSに連携し配車を実施

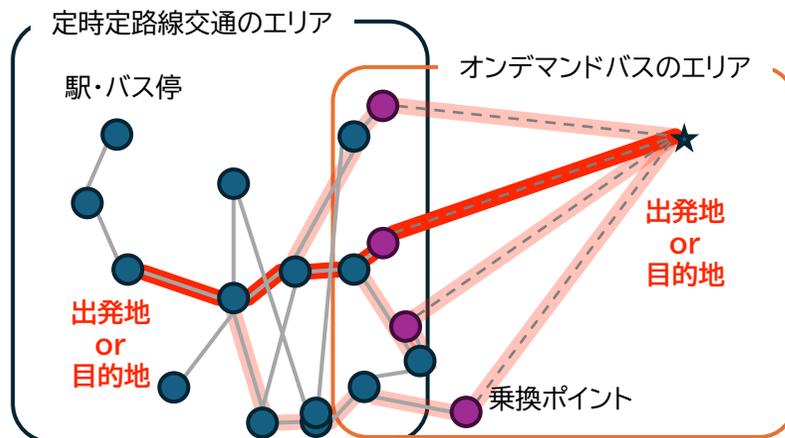
API

経路検索結果から配車予約可能



SAVSドライバーアプリ

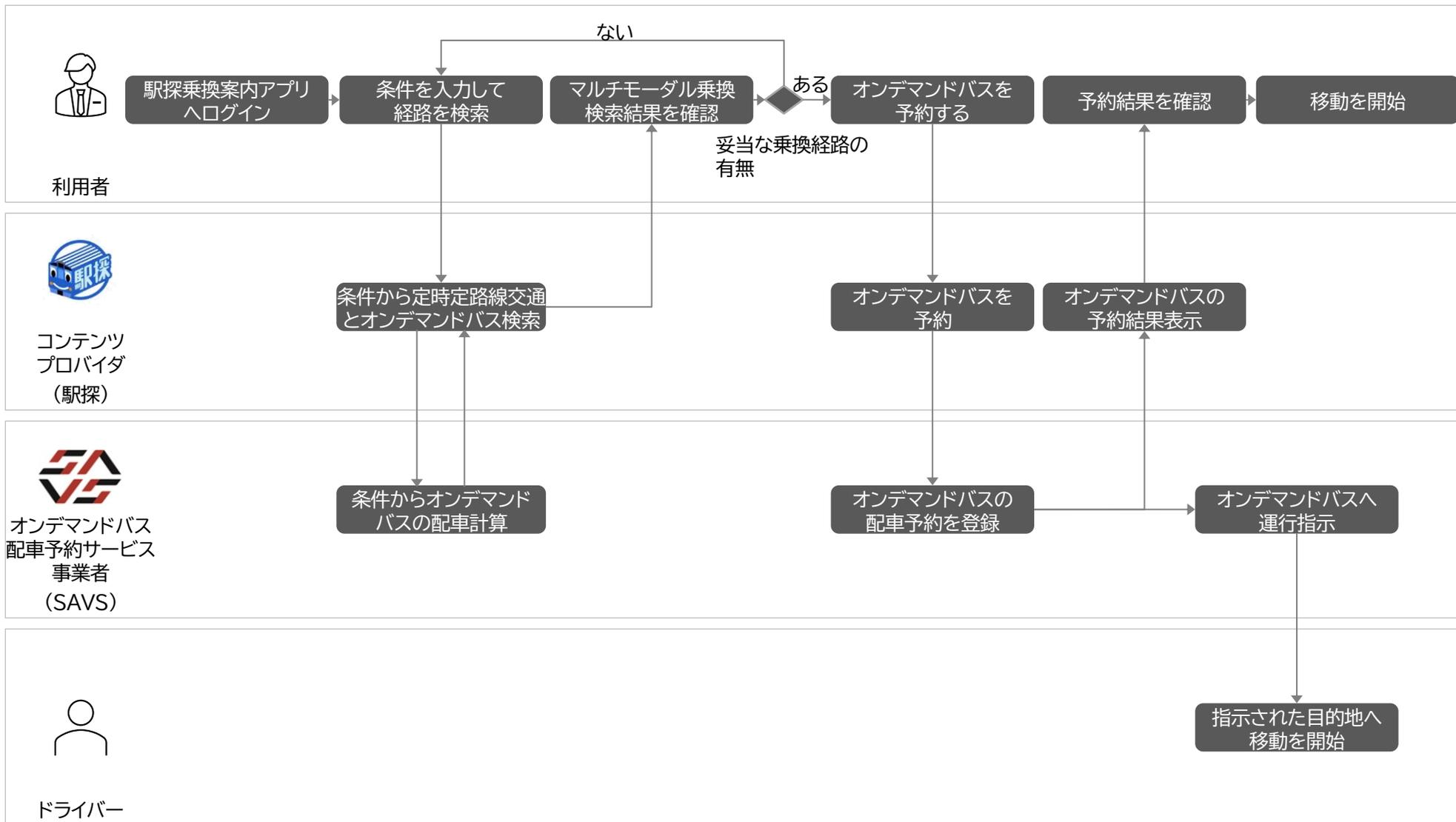
駅探アプリとSAVSアプリの連携



定時定路線とオンデマンドバスを跨いだ経路検索アルゴリズムのイメージ(ネットワーク図)

利用者が定時定路線交通とオンデマンドバスを横断した乗換経路を検索すると、検索結果画面から直接オンデマンドバスの配車を予約と配車を実現する

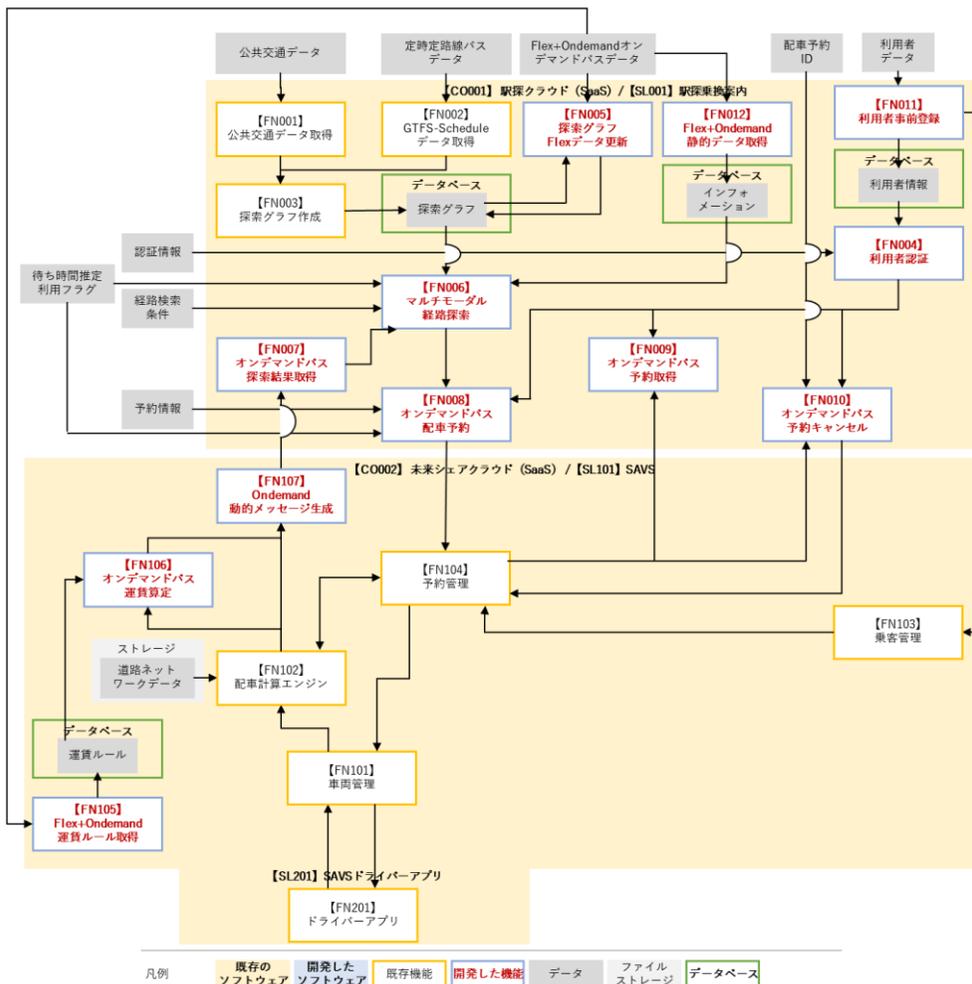
マルチモーダル乗換検索&リアルタイム配車予約の業務フロー



駅探乗換案内と未来シェアSAVS、SAVSドライバーアプリを基に、マルチモーダル乗換検索機能やリアルタイム配車予約機能を追加開発した

※詳細については(付録)GTFS-Flex/Ondemand活用システム システム設計書を参照
https://www.mlit.go.jp/commmons/tech_report/009/

システムアーキテクチャ図



システム機能一覧

ID	機能名	機能説明
FN001	公共交通データ取得	公共交通の路線や運賃、時刻表などのデータを提供している事業者から取得した公共交通データを取り込む。
FN002	GTFS-Scheduleデータ取得	GTFS Scheduleデータを取り込む。
FN003	探索グラフ作成	取り込んだ公共交通のデータに基づき、定時定路線交通の経路探索を行うためのグラフを構成する。
FN004	利用者認証	マルチモーダル乗換検索やオンデマンドバス配車予約を行う利用者の認証を行う。
FN005	探索グラフFlexデータ更新	GTFS-Flex及びGTFS-Ondemandの静的なデータを取り込み、オンデマンドバスの乗換ポイントとなるノードを探索グラフへ追加する。
FN006	マルチモーダル経路探索	入力された探索グラフ上で、マルチモーダルな乗換検索を実行する。
FN007	オンデマンドバス探索結果取得	条件に従って推定したオンデマンドバスの待ち時間と運賃を、GTFS-Ondemandのリアルタイムデータとして取り込む。
FN008	オンデマンドバス配車予約	検索結果から選択された経路と入力された予約情報を基に、オンデマンドバスの配車予約機能呼び出す。
FN009	オンデマンドバス予約取得	利用者の配車予約の一覧あるいは詳細を取得する。
FN010	オンデマンドバス予約キャンセル	乗車前の配車予約をキャンセルする。
FN011	利用者事前登録	実証実験開始前に、利用者の情報をデータベースへ直接登録する。
FN012	Flex+Ondemand静的データ取得	GTFS-Flex及びGTFS-Ondemandが拡張した静的なデータを取り込み、利用者へ提示すべき情報を取得する。
FN101	車両管理	車両情報を取得し配車計算エンジンへ与える。今後予定されている行き先をドライバーアプリへ配信する。
FN102	配車計算エンジン	道路ネットワークや車両の現在位置、今後の行き先などからオンデマンドバスの配車計算を行う。
FN103	乗客管理	配車予約の主体となる乗客を登録し管理する。
FN104	予約管理	リアルタイムな配車計算に基づき、配車予約を行う。
FN105	Flex+Ondemand運賃ルール取得	GTFS-Flex及びGTFS-Ondemandが拡張した静的なデータから運賃算定に必要なルールを取得する。
FN106	オンデマンドバス運賃算定	運賃ルールと配車計画に基づいてオンデマンドバスの運賃を算定する。
FN107	Ondemand動的メッセージ生成	配車計画データから出発地と到着地の待ち時間を推定し、算定した運賃と共にリアルタイムメッセージを組立て送信する。
FN201	ドライバーアプリ	現在位置や乗客の乗車状況などの車両情報を送信するとともに、今後予定されている行き先などをドライバーへ提示する。



駅探乗換案内及び未来シェアSAVSを中心に、GTFS-Ondemandのリアルタイムデータ扱ってモダンな乗換検索アプリのUI/UXを実現するためのOSSを利用した

利用した技術スタック

凡例

クラウド
サービス

ソフトウェア

ライブラリ・
フレームワーク

駅探乗換案内

SaaS



<https://ekitan.com/>

- ・(株)駅探社が運営する乗換案内サービス
- ・日本全国の定時定路線交通の乗換検索や時刻表表示等ができる

SAVS

SaaS



<https://www.miraishare.co.jp/>

- ・(株)未来シェア社が運営するオンデマンドバス配車予約サービス
- ・オンデマンドバスのリアルタイムな配車計算を行い、配車を予約することができる

PostgreSQL

データベース



<https://www.postgresql.org/>

- ・信頼性・堅牢性・パフォーマンスに優れたオープンソースのオブジェクトリレーショナルデータベース
- ・ユーザー定義型や関数、拡張モジュールによる機能拡張が容易

PostGIS

拡張モジュール



<https://postgis.net/>

- ・PostgreSQLデータベースに地理空間情報を格納し、操作するための地理空間拡張モジュール
- ・PostgreSQLに空間データ型、空間関数、空間インデックスなどを追加することで、地理情報システム(GIS)を実現するためのツールとして利用できる

gtfs-realtime-bindings

ライブラリ

<https://github.com/MobilityData/gtfs-realtime-bindings>

- ・GTFS Realtimeデータを扱うためのバインディングライブラリ
- ・GTFSの国際標準化団体であるMobilityDataが管理している

Express

フレームワーク



<https://expressjs.com/>

- ・バックエンドを構成するサーバーサイドフレームワーク
- ・堅牢で拡張性に富み、非同期処理を生かした高いパフォーマンスを誇る
- ・公開されているミドルウェアのエコシステムを活用することで、横断的機能を容易に組み込むことができる

Vue.js

フレームワーク



<https://vuejs.org/>

- ・駅探乗換案内UIのフロントエンドを構成するJavaScriptフレームワーク
- ・宣言的でリアクティブなコンポーネント指向のフレームワークであり、柔軟な画面設計と高いパフォーマンスを両立できる

FastAPI

フレームワーク



<https://fastapi.tiangolo.com>

- ・PythonでAPIを開発するためのフレームワーク
- ・Pythonの型ヒントを活用し、読みやすくメンテナンスしやすいコードを記述できる

駅乗換案内内のサービスUI/UXを基に、マルチモーダル乗換案内と、検索結果から直接オンデマンドバスのリアルタイム配車予約が可能なるUI/UXを開発した

UI/UXフロー

ログイン画面



ホーム画面



検索条件入力画面



出発地/目的地指定画面



駅・バス停指定画面



地図から指定画面



スポット指定画面



検索結果画面



配車予約確認画面



配車予約完了画面



利用者が直感的に利用できるよう、商用利用されているサービスを基にすることで、分かりやすさと操作しやすさを重視した画面を設計した

主に利用される画面のイメージ

ホーム画面



検索条件入力画面



出発地や到着地がオンデマンドバスの場合、stops.txtからオンデマンドバスの乗降スポット一覧を取得

trips.txtに定義されたluggage_allowed=1(荷物を持って乗車可)を反映

スポット指定画面



出発地や到着地がオンデマンドバスの場合、stops.txtから乗降スポットの緯度・経度を取得し地図上へ表示

- マルチモーダル乗換検索のホーム画面
- 既に配車予約が成立している場合、配車予約の概要が表示される

- 出発地や到着地、出発日時や到着日時、荷物の有無等の乗換検索条件を入力する画面
- 出発地や到着地は、駅・バス停指定画面や地図から指定画面及びスポット指定画面を用いて指定する

- GTFS-Flexに従って定義されたオンデマンドバスの乗降スポットを選択する画面
- 乗降スポットを指定すると、地図上でその位置を確認することができる(地図から指定画面も同様のUI)

利用者が直感的に利用できるよう、商用利用されているサービスを基にすることで、分かりやすさと操作しやすさを重視した画面を設計した

主に利用される画面のイメージ

検索結果画面

GTFS-Ondemandのリアルタイムメッセージが利用できない場合、乗換経路を検索する際にオンデマンドバスのリアルタイムの状況が把握できないため、到着直後に乗換可能と仮定した見なし乗換時間を表示

SAVSのリアルタイム配車計算で予測した乗換時間(GTFS-Ondemandのリアルタイムメッセージで取得)

SAVSのリアルタイム配車計算で予測した到着時間(GTFS-Ondemandのリアルタイムメッセージで取得)

GTFS-Ondemandのリアルタイムメッセージが利用できない場合、乗換経路を検索する際にオンデマンドバスのリアルタイムの状況が把握できないため、目的地までの直線距離から単純計算した到着時刻を表示



fare_leg_rules.txtで定義された固定運賃と fare_variable_rules.txtで定義された距離精算運賃から算出された変動運賃(GTFS-Ondemandのリアルタイムメッセージで取得)

booking_rules.txtで定義された予約タイプ

trips.txtで定義された重要な予約情報を表示

wait_rules.txtで定義された平均待ち時間

booking_rules.txtで定義された利用者向けメッセージ

- マルチモーダル乗換検索の検索結果を表示する画面
- 予約タイプや乗合可否など、オンデマンドバス特有の情報も併せて表示される
- 検索されたオンデマンドバスをこの画面から直接予約することができる

第4章 実証実験

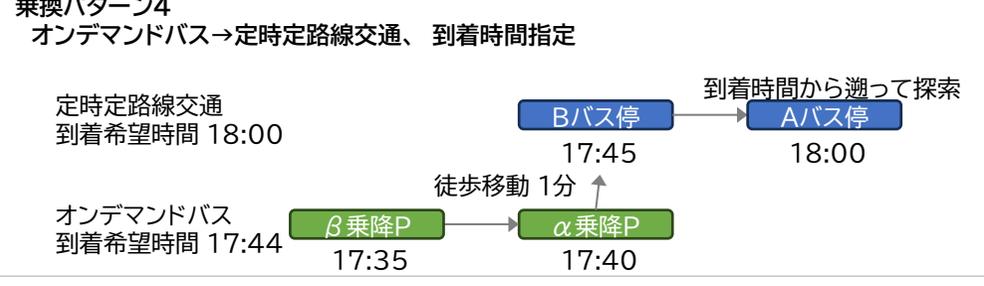
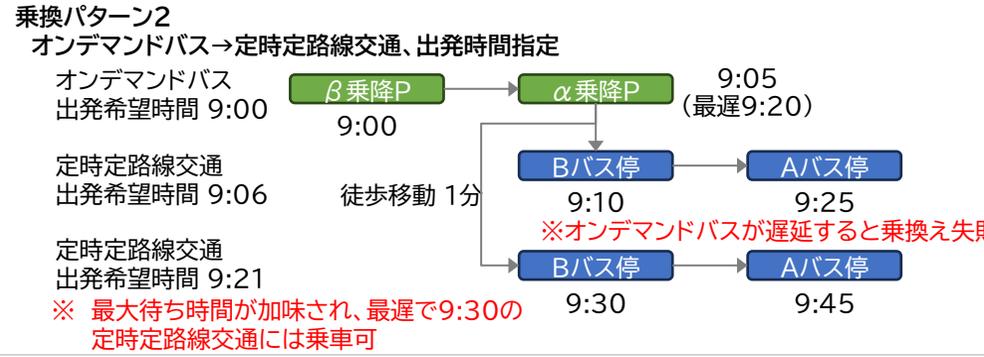
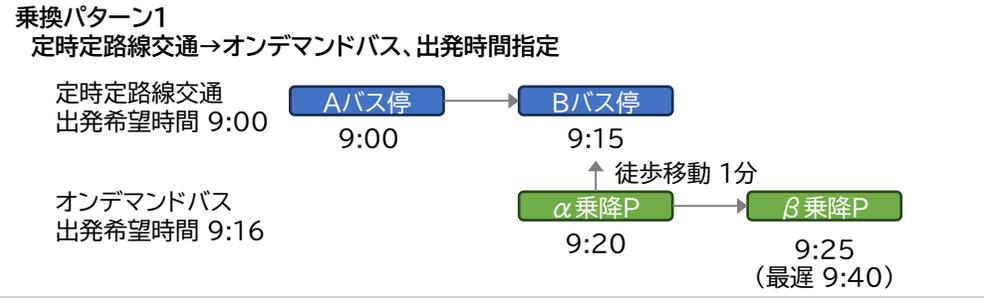
GTFS-Flex及びGTFS-Ondemandを活用したマルチモーダル乗換検索のベストプラクティスが創出され、オンデマンドバスの地域の足としての利便性が向上されることを目的に、北海道札幌市創成イーストエリアにて実証実験を行った。

その結果、9割を超える被験者から運行しているオンデマンドバスの発見性や利用性が向上したという評価が得られたとともに、今回取り組んだUI/UXの社会実装へも高い期待が寄せられていることが判明した。

机上検証と被験者によるサービス実証を通して、マルチモーダル乗換検索及びリアルタイム配車予約を用いることによるオンデマンドバスの利便性向上を評価した

実証メニュー一覧

実証メニュー	実施事項	被験者
机上実証: アルゴリズム 検証	<ul style="list-style-type: none"> マルチモーダル乗換検索のパターン(右図)に対して検索されるべき乗換経路の件数を想定し、想定どおりの結果が得られることをアルゴリズム単体で検証する 	-
サービス実証: データ作成工 数検証	<ul style="list-style-type: none"> GTFS-FlexとGTFS-Ondemandの仕様理解から、本プロジェクトでの実証実験で必要となる静的なオンデマンドバス用GTFSデータを作成するために必要な工数を検証する 	エンジニア (TIS1名、 駅探2名)
サービス実証: Zone-to- point型	<ul style="list-style-type: none"> Zone-to-point型の実証実験をTISが主体となって札幌市創成イーストエリアで実施する 被験者からアンケートを収集し、オンデマンドバスの利便性、利用への期待度を確認する 実証実験期間中に非機能要件(画面応答時間とシステム稼働率)も確認する 	TIS PJ関係者 (6名)
サービス実証: Point-to- zone型	<ul style="list-style-type: none"> Point-to-zone型の実証実験をTISが主体となって札幌市創成イーストエリアで実施する 被験者からアンケートを収集し、オンデマンドバスの利便性、利用への期待度を確認する 実証実験期間中に非機能要件(画面応答時間とシステム稼働率)も確認する 	札幌市民から 選んだモニ ター (22名)
サービス実証: Checkpoint 型	<ul style="list-style-type: none"> Checkpoint型の実証実験を札幌市が主体となって札幌市創成イーストエリアで実施する 被験者からアンケートを収集し、オンデマンドバスの利便性、利用への期待度を確認する 実証実験期間中に非機能要件(画面応答時間とシステム稼働率)も確認する 	札幌市民から 選んだモニ ター (22名)



「机上実証:アルゴリズム検証」におけるマルチモーダル乗換検索の4つのパターン



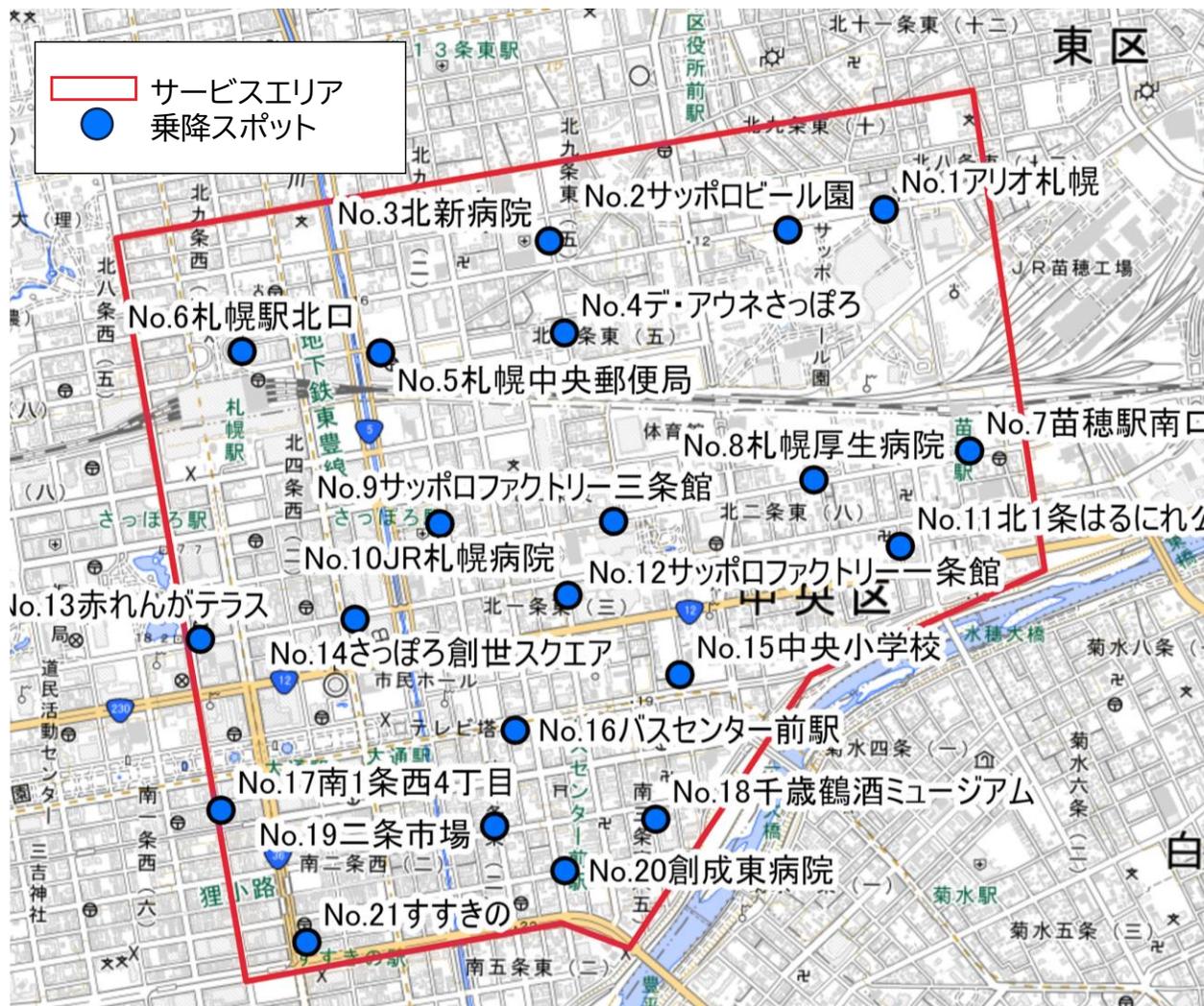
ビジネス価値、公共価値、ユーザー価値及び技術価値の四つの視点からKPIを策定、経路検索の満足度や社会実装への期待、システム性能や稼働率を定量的に評価した

検証仮説・検証項目・KPI

観点	検証仮説	検証項目	KPI
ビジネス価値	GTFS-Flex及びGTFS-Ondemandが定める標準化されたデータ仕様を活用することで、データ生成システムとデータ受領システム間のデータ送受信の実装が容易になり、またオンデマンドバスの静的なデータもGTFS-JPと同程度で容易に作成できる	GTFS-Ondemandを活用することによるシステム間連携機能の実装に必要な工数の削減割合	GTFS-Ondemandを活用したデータ配信機能実装における削減割合:50%
		GTFS-Flex及びGTFS-Ondemandの静的なデータを作成するための仕様理解からデータ作成までに必要な工数	GTFS-Flex及びGTFS-Ondemandの仕様理解から静的なデータを作成するために必要な工数:3人日
公共価値	地域専用の予約導線を介さない、一般的な乗換検索アプリを活用したマルチモーダル乗換検索・オンデマンドバス予約サービスが社会実装されることで、地域住民に加えて旅行者等の利用障壁が下がり、多様なユーザーによるオンデマンドバスの利用機会が増加する	ユーザーのサービス社会実装に対する期待度	利用への期待度:半数以上が期待 ※上記のほか、定性評価として、マルチモーダル経路検索の効果的な利用シーンを自由記述し確認
ユーザー価値	GTFS-Flex及びGTFS-Ondemandの活用により、一般的な乗換検索アプリから直接オンデマンドバスが予約可能となり、従来は地域専用の予約導線を介す必要のあった予約・利用のUXが向上する	オンデマンドバス予約の利便性の向上度合い	経路検索結果の納得度:半数以上が納得 オンデマンドバスの待ち時間の予実差:最大20分 オンデマンドバス待ち時間の納得度:半数以上が納得 サービス満足度:半数以上が満足
		GTFS-Flexのみに比べGTFS-Ondemand拡張仕様も利活用した場合のオンデマンドバスの利便性の向上度合い	利便性の向上度:半数以上が利便性が向上したと回答
技術価値	国際標準であるGTFS-Flex及びOndemandを活用し、国内で初めて実用に堪え得るマルチモーダル乗換検索システムを開発することで、その実装方法に関する技術ナレッジを社会に還元できる	マルチモーダル乗換検索として備えるべき機能要件	条件に合致するマルチモーダルな乗換経路が検索可能:テストケースごとに設定された件数が検索される
		マルチモーダル乗換検索として備えるべき非機能要件	ランディングページの応答時間:3秒以下(90パーセントایل値) 乗換検索結果ページの応答時間:7秒以下(90パーセントایل値) 実証実験期間中の稼働率(システム計画停止は含まず):95%/月

北海道札幌市の中心部であるが、公共交通がカバーしきれていない地域がある
創成イーストエリアを対象とした

北海道札幌市創成イーストエリア



本実証実験用に定めたエリアと乗降ポイント

No.	施設名	緯度	経度
1	アリオ札幌	43.0731	141.3713
2	サッポロビール園	43.0726	141.3683
3	北新病院	43.0723	141.3609
4	デ・アウネさっぽろ	43.0702	141.3614
5	札幌中央郵便局	43.0697	141.3557
6	札幌駅北口	43.0697	141.3514
7	苗穂駅南口	43.0675	141.3739
8	札幌厚生病院	43.0669	141.3692
9	サッポロファクトリー三条館	43.0659	141.3630
10	JR札幌病院	43.0658	141.3576
11	北1条はるにれ公園	43.0654	141.3719
12	サッポロファクトリー一一条館	43.0642	141.3616
13	赤れんがテラス	43.0631	141.3502
14	さっぽろ創成スクエア	43.0636	141.3550
15	中央小学校	43.0624	141.3651
16	バスセンター前駅	43.0611	141.3600
17	南1条西4丁目	43.0592	141.3509
18	千歳鶴酒ミュージアム	43.0591	141.3644
19	二条市場	43.0589	141.3594
20	創成東病院	43.0579	141.3616
21	すすきの	43.0562	141.3536

乗降ポイント一覧

サービス実証



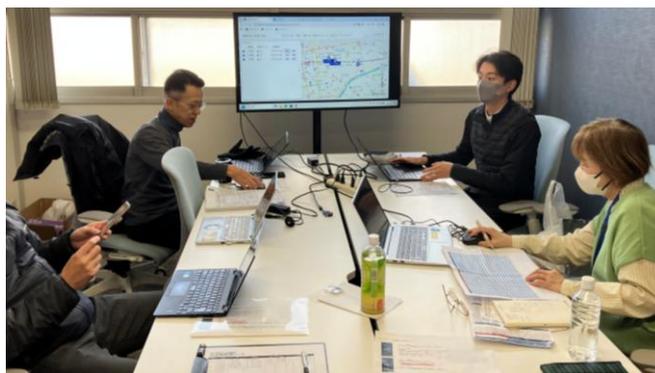
利用者がオンデマンドバスに乗車する様子



オンデマンドバスドライバーの様子



オンデマンドバス運転席とドライバーアプリ



オンデマンドバス管理センター



実車での乗車モニターへの説明



実車でのドライバーへの説明

GTFSが定める国際標準仕様を活用することで、システム間のデータ連携を容易に実現でき、オンデマンドバスのデータも容易に定義することができる

結果のまとめ

検証仮説

GTFS-Flex及びGTFS-Ondemandが定める標準化されたデータ仕様を活用することで、データ生成システムと受領システム間での送受信の実装が容易になり、またオンデマンドバスの静的なデータもGTFS-JPと同程度で容易に作成できる。

検証結果

本プロジェクトでは、駅探乗換案内上でマルチモーダル乗換検索を実行する際に、SAVSから駅探乗換案内へ配車可能なオンデマンドバスの出発地と目的地の待ち時間及び運賃の情報がリアルタイムに配信されるが、GTFS-Ondemandにて標準化されたリアルタイムメッセージのデータ仕様を活用することで、リアルタイムデータのシステム間送受信を容易に実現できた。具体的には、GTFSの標準仕様を利用せずに同様のシステム間連携を行ったSAVS過去事例と比較し、仕様調整から単体テストまで50%削減できたと見込まれる。

また、GTFS Schedule及びGTFS Realtimeの仕様を理解しているが実務で取り扱ったことがないエンジニアと、GTFS Scheduleデータを実務で取り扱っているエンジニア2名により、オンデマンドバスの静的なデータ作成とチェックを行った。Mobility Dataが公開するデータ項目仕様やベストプラクティス、サンプルデータを熟読しデータ項目仕様と対応付け、ファイル間の主キーの関係を把握することで、オンデマンドバス特有のデータ項目も円滑に定義することができた。

得られた示唆

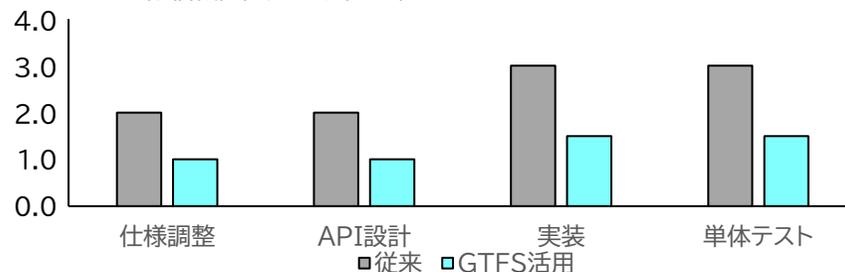
GTFS-Ondemandの国際標準仕様の活用による工数削減

GTFS-Ondemandの国際標準仕様を活用することで、プロジェクト独自にデータ連携仕様を定める場合と比較して、仕様の調整やAPIの設計に要する工数を削減できることにより50%程度工数が削減できることが分かった。

GTFS-Flex及びGTFS-Ondemandの静的なデータの作成工数

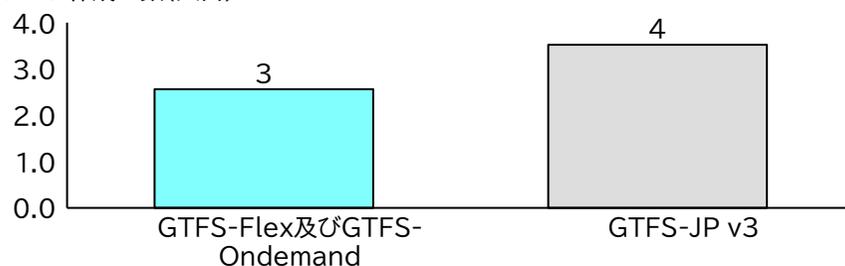
GTFS ScheduleとGTFS Realtimeの基本的な知識を持つITエンジニアであれば、実際のオンデマンドバスの運行要件に従って、GTFS-Flex及びGTFS-Ondemandを用いたオンデマンドバスの静的なデータ作成とチェックを3人日程度で実施できることが分かった。同程度のスキルと経験を持ったエンジニアが、静的バス情報フォーマット(GTFS-JP)仕様書[第3版]に従った路線バスの静的なデータを3.5人日程度で作成できたため、同程度の工数と見なすことができる。

リアルタイムデータ配信機能開発工数(人日)



GTFS-Ondemandリアルタイムデータ配信機能の開発工数の比較

静的データ作成工数(人日)



GTFSの静的データ作成工数の比較

```
booking_rule_id,booking_type,message,phone_number,info_url,booking_url,download_app_url
rule_001,0,実証実験の為、運賃は表示のみで受け取らな
い,000-XXXX-XXXX,https://maas.ekitan.com/sapporo-pilot/web/,https://maas.ekitan.com/sapporo-pilot/web/,https://maas.ekitan.com/sapporo-pilot/download/
```

booking_rules.txtの例

```
wait_rule_id,stop_id,service_id,start_time,end_time,mean_wait_time,safe_wait_time,max_wait_time
wait_001,lg_001,c_001,09:00:00,16:00:00,15,15,30
wait_002,lg_002,c_001,09:00:00,16:00:00,15,15,30
wait_003,area_001,c_001,09:00:00,16:00:00,15,15,30
```

wait_rules.txtの例

```
trip_id,location_id,location_group_id,stop_sequence,start_pickup_drop_off_window,end_pickup_drop_off_w
indow,pickup_type,drop_off_type,pickup_booking_rule_id,drop_off_booking_rule_id,pickup_proximity_level,drop_off_proximity_level,continuous_pickup,continuous_drop_off,mean_wait_time,safe_wait_time,max_wait_time,wait_rule_id
r001_c001_001,,lg_002,1,09:00:00,21:00:00,2,1,rule_001,rule_001,1,1,1,1,15,15,30,wait_002
(以下略)
```

stop.times.txtの例

```
{
  "header": {
    "gtfs_realtime_version": "2.0",
    "timestamp": 1748738603
  },
  "entity": {
    "id": "001",
    "on_demand": [
      {
        "wait_time_update": {
          "wait_location": "pickup_0001",
          "trip_id": "trip_001",
          "wait_time": 612,
          "max_wait_time": 1512
        }
      },
      {
        "fare_update": {
          "fare_leg_id": "fare_leg_001",
          "origin": "pickup_0001",
          "destination": "dropOff_0021",
          "amount": 100.0
        }
      }
    ]
  }
}
```

GTFS-Ondemandリアルタイムメッセージの例



検証方法

KPI詳細

KPI	定義	目標設定根拠
GTFS-Flex及びGTFS-Ondemandの仕様理解から静的なデータを作成するために必要な工数:3人日	各タスクの開始時間と終了時間の差分(分単位)を積算	静的バス情報フォーマット(GTFS-JP)仕様書[第3版]に従って路線バスの静的データを作成する工数と同程度と見積

KPIの計測方法

- エンジニア1名が実証で用いるオンデマンドバスの静的データ作成業務を行い、その静的ファイルの整合性を駅探のエンジニア1名がチェックし、実工数を計測した。
- 作業工数を比較するために、オンデマンドバスの静的データ作成者とは別のエンジニア1名が実証エリアで運行する路線バス(1路線)の静的データ作成業務を行い、駅探エンジニア1名がその静的ファイルの整合性をチェックして、実工数を計測した。
- 具体的な検証フローは以下の通り
 - GTFS-Flexの仕様とベストプラクティス及びGTFS-Ondemandの仕様案を確認し、どのような項目を定義すべきかを把握する。
 - GTFS-Flex及びGTFS-Ondemandが定めるデータ項目を全て網羅できるように、実証要件とデータ項目の対応付け。
 - GTFS-Flex及びGTFS-Ondemandが定めるファイルごとに、以下のタスクを実施する。
 - 作業開始日時を記録する。
 - データ項目の対応付けに従い、GTFS-Flex及びGTFS-Ondemandが定めるファイルを作成する。
 - 作業終了日時を記録する。
 - 静的データファイル一式の整合性をチェックする。
 - 作業工数を集計する。

結果の詳細

静的なGTFSデータ作成工数の比較

- GTFS-Flex及びGTFS-Ondemandに従った場合(オンデマンドバス)

タスク	工数(時間)
GTFS-Flex仕様確認	4.0
GTFS-Ondemand仕様確認	5.0
実証要件とGTFS-Flex及びGTFS-Ondemandの項目定義との対応付け	1.0
ファイル作成	
agency.txt	0.1
booking_deep_links.txt	0.2
booking_rules.txt	0.6
calendar.txt	0.1
fare_leg_rules.txt	0.5
fare_products.txt	0.5
fare_variable_rules.txt	0.5
location_group_stops.txt	0.3
location_groups.txt	0.3
locations.geojson	0.5
routes.txt	0.3
stop_times.txt	0.6
stops.txt	1.0
trips.txt	0.5
wait_rules.txt	0.5
整合性チェック	4.0
定義ファイル一式の作成工数	20.5

- 静的バス情報フォーマット(GTFS-JP)仕様書[第3版]に従った場合(路線バス)

タスク	工数(時間)
既存データ項目と静的バス情報フォーマット(GTFS-JP)[第3版]との対応付け	5.0
ファイル作成	
agency.txt	0.1
calendar.txt	0.75
calendar_dates.txt	1.25
fare_attributes.txt	0.5
fare_rules.txt	0.5
feed_info.txt	0.1
routes.txt	5.0
stop_times.txt	3.0
stops.txt	3.0
translations.txt	4.0
trips.txt	1.0
整合性チェック	4.0
定義ファイル一式の作成工数総工数	28.2



本プロジェクトにより、オンデマンドバスの検索や利用が容易になると評価され、本プロジェクトによるオンデマンドバスの利用機会増大が期待できる

結果のまとめ

検証仮説

地域専用の予約導線を介さない、一般的な乗換検索アプリを活用したマルチモーダル乗換検索・オンデマンドバス予約サービスが社会実装されることで、地域住民に加えて旅行者等の利用障壁が下がり、多様なユーザーによるオンデマンドバスの利用機会が増加する。

検証結果

本プロジェクトでは、定時定路線交通とオンデマンドバスが跨がるマルチモーダルな乗換検索とシームレスでリアルタイムな配車予約と移動によって、オンデマンドバスが利用しやすくなるか、社会実装された場合に実際に利用しやすくなるかを評価した。具体的には、本プロジェクトのUI/UXを利用者に体験してもらい、移動経路に沿った適切なオンデマンドバスを見つけやすくなったか、オンデマンドバスが利用しやすくなったかを5段階評価のアンケートにより確認した。

その結果、オンデマンドバスの利便性向上に関して、9割以上の被験者が「非常にそう思う」「そう思う」と回答する等非常に高い評価が得られた。また、同様に札幌市での社会実装への期待度についても、9割以上の被験者が仕様したいと思うと回答する等社会実装へ高い期待があることが確認できた。これは、従来、定時定路線交通とオンデマンドバスをまたがる乗換検索が可能なサービスがなく、予約に地域専用導線等を介する必要があるため使いにくい現状に対し、一般的な経路検索アプリを介して乗換検索と配車予約が一气通貫で可能となったことが評価されたと考えられる。

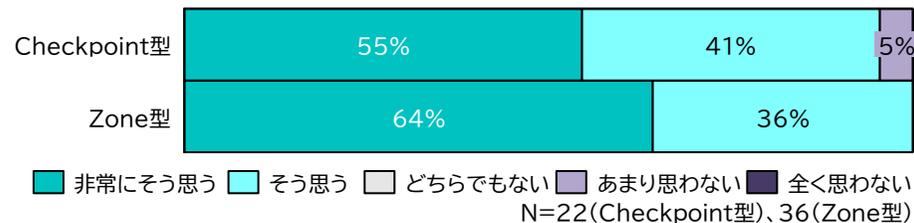
得られた示唆

リアルタイムな経路検索・配車予約実現によるオンデマンドバスの利便性向上

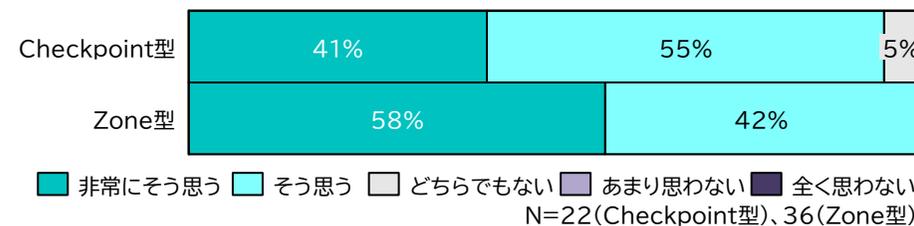
定時定路線交通とオンデマンドバスにまたがった乗換経路を、オンデマンドバスのリアルタイムな情報を加味して検索・配車予約ができる本プロジェクトは、オンデマンドバスの利用性を大きく向上させ得ることが分かった。

今までこの2つの公共交通機関(定時定路線交通とオンデマンドバス)は、利用者が乗換経路を考え、検索し配車予約する必要があったが、本プロジェクトにより手間が削減され、オンデマンドバスの利用障壁が低下したためと考えられる。

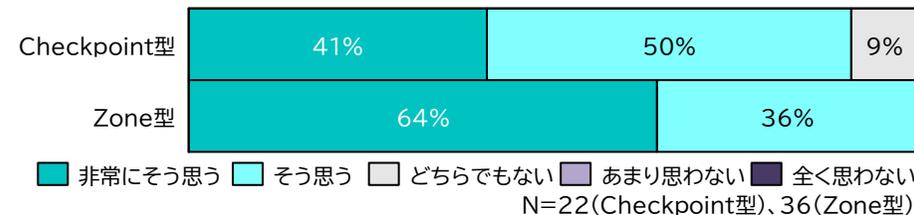
また、実サービスでの利用への期待度も高く、本プロジェクトによりオンデマンドバスの利用機会増大にも貢献できることが確認できたため、今後の社会実装に向けた着実な展開が重要となると考えられる。



質問. オンデマンドバスを見つけやすくなったと感じられましたか？



質問. オンデマンドバスが利用しやすくなったと感じられましたか？



質問. 本実証実験で体験したサービスが札幌市で実現したら利用したいと思いますか？

被験者
コメント抜粋

- ・ 定時定路線交通やオンデマンドバスの利用がシームレスに活用できるサービスになり、意識せず利用できるモビリティサービスになれば、生活圏での利用ができる。
- ・ オンデマンドバスの頼み方が不明なので、オンデマンドバスの利用方法がサービス内で統合されていて良い。
- ・ 札幌は積雪時に特に歩くことがおっくうなため、オンデマンドバスを含めた経路提案は非常に良いと感じた。
- ・ 不慣れな場所でも目的地に簡単に到着でき便利だと思う。

質問. 本実証実験で体験したサービスの利便性についてコメントはありますか？



検証方法

KPI詳細

KPI	定義	目標設定根拠
利用への期待度 :半数以上が期待	5段階のリッカート尺度による被験者へのアンケート	社会実装する価値があることを定性的に捉えるため

KPIの計測方法

- Checkpoint型の実証では、札幌市民22名が定時定路線交通とオンデマンドバスとにまたがったマルチモーダルな乗換検索とシームレスなオンデマンドバス配車予約を体験し、実際の乗車体験の後にアンケートに回答した。
- Zone型の実証では、TIS関係者7名が定時定路線交通とオンデマンドバスとにまたがったマルチモーダルな乗換検索とシームレスなオンデマンドバス配車予約を体験し、実際の乗車体験の後にアンケートに回答した(複数回の体験をした被験者は複数回答し、のべ36名)。
- 具体的な検証フローは以下の通り
 - 被験者は事前に計画された出発地と目的地、乗車時間(あるいは降車時間)、ログインアカウントを確認する。
 - 被験者は自身のスマートフォンで駅探乗換案内アプリを起動し、指定されたアカウントでログインする。
 - 被験者は指定された条件に従ってマルチモーダルな乗換経路を検索し、提案された乗換経路を確認して乗換経路を一つ選択し、オンデマンドバスを予約する。
 - 被験者は選択した乗換経路に沿って乗車し、定時定路線交通とオンデマンドバスの乗り換えも体験する。
 - 被験者は移動が完了した後にアンケートに回答する。
- 実証終了後にアンケート結果を集計し、次の5つの質問項目から利用への期待度を算出する。

質問項目

設問	質問項目
1	オンデマンドバスを見つけやすくなったと感じられましたか?(5段階評価)
2	オンデマンドバスが利用しやすくなったと感じられましたか?(5段階評価)
3	本実証実験で体験したサービスが札幌市で実現したら利用したいと思いますか?(5段階評価)
4	本プロジェクトへの将来的な期待について、ご意見やご感想をお書きください。(自由回答)
5	本プロジェクトの効果的な利用シーンについて、ご意見やご感想をお書きください。(自由回答)

被験者一覧

#	実証	被験者の年代	被験者の性別	被験者数
1	Checkpoint型	20代	男性	5名
		20代	女性	1名
		30代	男性	2名
		30代	女性	2名
		40代	男性	3名
		40代	女性	3名
		50代	男性	3名
		50代	女性	2名
		60代以上	男性	1名
		2	Zone型	20代
20代	女性			1名
30代	男性			2名
30代	女性			6名
40代	男性			3名
40代	女性			4名
50代	男性			9名
50代	女性			2名
	60代以上	男性	4名	



結果の詳細

公共価値のアンケート結果

- アンケートの結果、本プロジェクトはオンデマンドバスの発見しやすさと利用しやすさを大きく向上させ、社会実装へも高い期待が持たれていることがわかった。

市民



40代 女性

定時制路線が通っていないエリアでも、自家用車がなくても行きやすくなり、行動範囲が広がる。住宅街など公共交通の届かない場所で効果的に活用できる。



40代 女性

スマートフォンに慣れていない高齢者でも、利用しやすく分かりやすいサービスであってほしい。



50代 男性

定時制路線と異なり、通院などのシーンでは乗降スポットまでの距離が短くなり助かる。郊外における高齢者や通院される方は特にニーズが高いと思われる。

- その他、アンケートの自由回答の全回答は以下のとおりである。

5段階評価の結果

質問1: オンデマンドバスを見つけやすくなったと感じられましたか?

実証	回答数				
	非常にそう思う	そう思う	どちらでもない	あまり思わない	全く思わない
Checkpoint型 (N=22)	12	9	0	1	0
Zone型(N=36)	23	13	0	0	0

質問2: オンデマンドバスが利用しやすくなったと感じられましたか?

実証	回答数				
	非常にそう思う	そう思う	どちらでもない	あまり思わない	全く思わない
Checkpoint型 (N=22)	9	12	1	0	0
Zone型(N=36)	21	15	0	0	0

質問3: 本実証実験で体験したサービスが札幌市で実現したら利用したいと思いますか?

実証	回答数				
	非常にそう思う	そう思う	どちらでもない	あまり思わない	全く思わない
Checkpoint型 (N=22)	9	11	2	0	0
Zone型(N=36)	23	13	0	0	0

・その他、アンケートの自由回答の全回答は以下のとおりである。

質問4:本プロジェクトへの将来的な期待について、ご意見やご感想をお書きください。

年齢	性別	回答
20代	女性	普段、札幌駅周辺でバスを使ったことがないため、移動の選択肢にないのと、〇〇直行など直接行きたい場所に向かってくれるバスが現時点であるため利用しやすいかといわれるとしづらと思う。アプリをつくる、地下鉄や市電、その他の交通機関より安いなど、他社と比べて魅力がある場合は利用価値が上がると思う。SAPICAのようにポイント還元がある、利用するとお店で割引、クーポンが使えるなど。
30代	女性	非常に期待しています。特に札幌のような雪の降るエリアや、高齢者が多いエリアにあるとよいと思います。
30代	女性	どこでもいつでも呼べる。安い。
50代	女性	もう少し移動距離が長かったら良いかなと思った。
50代	女性	停車位置が分かる全体マップなどが予め見られたら予定が立ちやすいと思う
50代	男性	期待度は高い。観光地が離れているエリアを対象とするか、住民向けに特化したほうがよい。
50代	男性	観光に行ったときに使いたい。
60代以上	男性	公共交通機関がない現状、早期の実現を求む
60代以上	男性	苗穂付近は新興住宅地もあり、駅からの距離が近いマンション・住宅が多いので大変有効だと思う。
60代以上	男性	毎日利用、または利用頻度が高い人が発生するかもしれないと思った。

質問5:本プロジェクトの効果的な利用シーンについて、ご意見やご感想をお書きください。

年齢	性別	回答
20代	男性	休日の買い物や観光での、市内を巡る時。
20代	男性	時間に余裕がある際のちょっとしたお出かけ。地下歩道が遠い住民が悪天候時に駅に足を運ぶ手段として。
20代	男性	公共交通機関が少ない地域や複数の乗換が発生する地域間の移動の際には効果的であると思う。また、自分の居住地から比較的近いスポットから乗降できるため、高齢者等にとっては便利な機能になると考える。
20代	男性	公共交通の届かない場所(住宅街等)で効果的に使用されると考える
30代	女性	住民だったらぜひ利用したい。利用登録が簡易であれば観光でも利用したい。
30代	女性	荷物等が多くて満員の電車や地下鉄に乗りたくない時、駅やバス停から離れた場所に目的地がある時
30代	男性	時間に縛りのない余裕がある際の移動
30代	男性	地下鉄やJRなどの公共交通からすこし離れた場所へ行きたいときに使いたい。冬などは特に自宅から最寄りの駅まで使えると便利。
40代	女性	旅先など不慣れな場所で目的地に着くと便利だと思います。
40代	女性	観光客向けのサービスとして活用できると思います。加えて、札幌在住者向け固定のバスなどがあると利用しやすくなるように思いました。
40代	女性	買い物やレジャー、時間に余裕があるけど楽に移動したいとき
40代	女性	JRも地下鉄も路線バスも通っていない場所にも自家用車がなくても行きやすくなって行動範囲が広がる。苗穂駅方面から市電に乗り継ぎたい場合など、乗り換えがないなら多少時間がかかってもデマンド交通を利用したいと思った。(天気が悪い、荷物が多い場合などは特に)
40代	男性	事務所から街中へのアクセス
40代	男性	日常使いが優先されつつ、キャパシティに余裕がある場合は観光客も利用するといった形が望ましいと考える
50代	女性	定時定路線路線での乗り継ぎが億劫と感じる場所間の移動、行きたい場所があるがどの経路で行くのが最適かが分からない時など
50代	男性	京都市のように、バスが多くて、観光地がない地域にフィットしそう。観光というより住民だったら使いたい。高齢者や悪天候時に特によさそう。
50代	男性	郊外における高齢者や通院される方は特にニーズが高いと思われる
50代	男性	通勤通学(バス便数減少により影響を受けている為)
50代	男性	通院や買い物など時間にゆとりがある移動には便利。通勤、通学等時間に余裕を持たない移動については利用しづらい。用途を明確にして、また徒歩が発生するにしてもスポット数を限定したほうが結果的に便利になるような気がする
60代以上	男性	定時定路線やデマンドもしくは通常のタクシー等の利用がシームレスに利活用できるサービスになり意識せず利用できるモビリティサービスになれば、特に高齢者や子供連れでの生活圏での利用ができると良い。また観光利用にて海外客も含めた利用ができると良い。
60代以上	男性	降雪時は特に便利だと思う。高齢者にとっては最適と思うが、時間・料金・乗り換えの回数人がシーンにより違うため、その都度ニーズに対応できてよいと思った。
60代以上	男性	予め駐車場所が分かっていると利用しやすいと感じた



オンデマンドバスを利用する際のUI/UXには一定の評価を得たものの、社会実装に向けた更なるUI/UX改善とチューニングが必要であることが明らかとなった

結果のまとめ

検証仮説

GTFS-Flex及びGTFS-Ondemandの活用により、一般的な乗換検索アプリから直接オンデマンドバスが予約可能となり、従来は地域専用の予約導線を介す必要のあった予約・利用のUXが向上する。

検証結果

最適な乗換経路の提案について、本プロジェクトで開発したマルチモーダルな乗換経路検索アルゴリズムにより、Checkpoint型では6割強程度、Zone型では全ての被験者から納得のいく乗換経路を検索できたという評価が得られた。

一方で、納得がいかない乗換経路が提案されたケースもあった。原因としては、オンデマンドバスではなく徒歩で移動する方が有利と判定する場合がある、オンデマンドバスが少し遅れると乗換えに失敗するリスクのある経路を提案する等の要因が考えられる。

予測待ち時間の精度について、利用者の実際の待ち時間と、検索時点での予測待ち時間との差異は、平均1.7分程度に収まった。

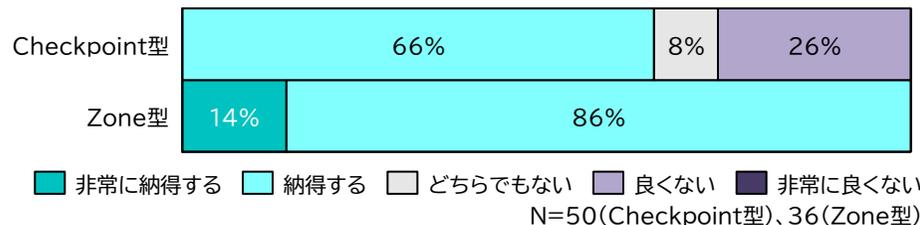
一方で、乗り換えが間に合わなかったケースが6件報告された。これは、乗換スポットと駅・バス停間との徒歩時間が直線距離から一律で設定され、札幌駅北口等での乗換所要時間が実際より短く算出されていたためである。

利用者のUI/UXの更なる向上について、改善を実感した利用者は4割に留まった。これは、例えば30分や1時間先の乗車希望日時で乗換検索をした場合、GTFS-Ondemandを用いず、リアルタイム情報を一切考慮せずに検索をした乗換経路であっても、そのとおりにオンデマンドバスが配車された場合があったためだと考えられる。利用者の移動体験がGTFS-Ondemand拡張仕様の使用有無に影響されないよう、よりケースを絞り込んだ再評価が必要になると考えられる。

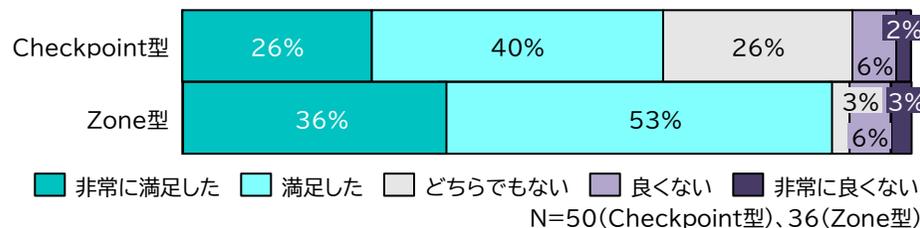
得られた示唆

UI/UXの更なる改善の必要性

利用者の満足度をより向上させ「実際に使ってもらえるアプリ」にするためには、オンデマンドバスの乗換えに失敗しないように、後続の定時定路線交通とつなぎ込むアルゴリズムの調整や、UI/UXの更なる改善が必要であることが分かった。また、鉄道や定時定路線バスとの確実な乗換のためには、乗換スポットの設定は実際の徒歩での所要時間を踏まえる必要があることが分かった。



質問. オンデマンドバスを含む適切な乗換経路を提案したいと思いますか？



質問. 乗換待ち時間についての満足度は？



質問. GTFS-Flexに加えGTFS-Ondemandも利用した方が利便性が良くなったと思いますか？

被験者コメント抜粋

- ・ 徒歩で行った方が近いのに別の経路が提示される場合や、目の前の公共交通に乗らないで徒歩のルートが提案されることがあった。
- ・ 遅延が発生する場合や早着の可能性がある場合は、プッシュ通知などがあると良い。

質問. 本実証実験で体験したサービスの利便性についてコメントはありますか？

検証方法

KPI詳細

KPI	定義	目標設定根拠
経路検索結果の納得度 :半数以上が納得	5段階のリッカート尺度による被験者へのアンケート	マルチモーダル乗換案内アルゴリズムが適切な乗換経路を検索できることを定性的に捉えるため
オンデマンドバスの待ち時間の予実差 :最大20分	被験者へのアンケートにて得た待ち時間予実差の最大値	マルチモーダル乗換案内アルゴリズムが適切な乗換経路を検索できることを定性的に捉えるため
オンデマンドバス待ち時間の納得度 :半数以上が納得	5段階のリッカート尺度による被験者へのアンケート	マルチモーダル乗換案内アルゴリズムが適切な乗換経路を検索できることを定性的に捉えるため
GTFS-Flexのみに比べGTFS-Ondemandも活用した場合の利便性の向上度 :半数以上が利便性向上	5段階のリッカート尺度による被験者へのアンケート	GTFS-Ondemandのリアルタイムメッセージを用いてリアルタイムな配車計算結果を活用したほうが、より適切な乗換経路を検索できることを定性的に捉えるため

KPIの計測方法

- Checkpoint型の実証では、札幌市民22名が定時定路線交通とオンデマンドバスとにまたがったマルチモーダルな乗換検索とシームレスなオンデマンドバス配車予約を体験し、実際の乗車体験の後にアンケートに回答した(複数回の体験をした被験者は複数回答し、のべ50名)。
- Zone型の実証では、TIS関係者7名が定時定路線交通、オンデマンドバスに跨ったマルチモーダルな乗換検索とシームレスなオンデマンドバス配車予約を体験し、乗車の後にアンケートに回答した(複数回の体験をした被験者は複数回答し、のべ36名)。
- 被験者の一部(Checkpoint型のべ14名、Zone型のべ15名)は、GTFS-FlexとGTFS-Ondemandを双方活用した通常の乗車体験に加え、GTFS-Ondemandのリアルタイム情報を利用しないモードに設定したシステム上で、通常の乗車体験と同一の出発地・目的地・出発時間(あるいは到着時間)でマルチモーダル乗換検索とシームレスなオンデマンドバス配車予約を体験し、両者を比較したアンケートに回答した。

- 具体的な検証フローは以下の通り
 - 被験者は事前に計画された出発地と目的地、乗車時間(あるいは降車時間)、ログインアカウントを確認する。
 - 被験者は自身のスマートフォンで駅探乗換案内アプリを起動し、指定されたアカウントでログインする。
 - 被験者は指定された条件に従いマルチモーダルな乗換経路を検索し、提案された乗換経路を確認して乗換経路を一つ選択し、オンデマンドバスを予約する。
 - 被験者は選択した乗換経路に沿って乗車し、定時定路線交通とオンデマンドバスの乗り換えも体験する。
 - 被験者は移動が完了した後にアンケートに回答する。
- 実証終了後にアンケート結果を集計し、次の6つの質問項目から乗換経路や待ち時間の納得度を算出した。

質問項目

設問	質問項目
1	オンデマンドバスを含む適切な乗換経路を提案したと思いますか?(5段階評価)
2	乗換待ち時間についての満足度は?(5段階評価)
3	電車や路線バス等の公共交通とオンデマンドバスを乗り換えた際に、乗換案内アプリが表示していた時間とどの程度のズレがありましたか?(分単位の数値で回答)
4	乗換案内アプリが提案した経路について、ご意見やご感想をお書きください。(自由回答)
5	乗換待ち時間について、ご意見やご感想をお書きください。(自由回答)
6	本プロジェクトへの全体的な満足度について、ご意見やご感想をお書きください。(自由回答)

- GTFS-Ondemandリアルタイムメッセージを用いる通常のシステムと、GTFS-Ondemandリアルタイムメッセージを用いないシステムとの双方を体験した15名には、実証終了後のアンケートにて次の追加質問も行い、GTFS-Ondemandのリアルタイムメッセージを用いることによる利便性の向上度を算出した。

追加質問項目

設問	質問項目
1	GTFS-Flexに加えGTFS-Ondemandも利用した方が利便性が良くなったと思いますか?(5段階評価)
2	GTFS-Flexに加えGTFS-Ondemandも利用したことによる利便性乗降について、ご意見やご感想をお書きください。(自由回答)



検証方法

被験者一覧

# 実証	被験者の年代	被験者の性別	被験者数
1 Checkpoint型	20代	男性	12名
	20代	女性	3名
	30代	男性	5名
	30代	女性	2名
	40代	男性	3名
	40代	女性	10名
	50代	男性	8名
	50代	女性	4名
	60代以上	男性	3名
	2 Zone型	20代	男性
20代	女性	1名	
30代	男性	2名	
30代	女性	6名	
40代	男性	3名	
40代	女性	4名	
50代	男性	9名	
50代	女性	2名	
60代以上	男性	4名	

GTFS-Ondemandリアルタイムメッセージが無いシステムも体験した被験者一覧

# 実証	被験者の年代	被験者の性別	被験者数
1 Checkpoint型	20代	男性	4名
	20代	女性	0名
	30代	男性	2名
	30代	女性	1名
	40代	男性	2名
	40代	女性	2名
	50代	男性	2名
	50代	女性	0名
	60代以上	男性	1名
	2 Zone型	20代	男性
20代	女性	0名	
30代	男性	2名	
30代	女性	2名	
40代	男性	2名	
40代	女性	2名	
50代	男性	2名	
50代	女性	0名	
60代以上	男性	1名	



結果の詳細

ユーザー価値のアンケート結果

- アンケートの結果、利用者の満足度をより向上させ「実際に使ってもらえるアプリ」にするためには、アルゴリズムのチューニングやUI/UXの更なる改善が必要であることが分かった。

市民
20代 女性

バスも何度か遅れてきたので、冬など天候が悪いときの遅延が少なくなればより良くなると思う。遅延が発生する場合や早着の可能性のある場合は、プッシュ通知等があると良い。

40代 女性

今回は実証実験でしたが、このバスに乗ってよいのか？ この場所でよいのか？ と不安に感じたので、見せ方や利用方法の説明、訴求、PRが重要であると感じました。

20代 男性

GTFS-Ondemandも併用した方が、時刻の目安がイメージしやすい。GTFS-Flexのみでは乗り継ぎに間に合わない経路が予約された。

60代 男性

車両の到着時間が表示された時間より早かったため適切な時刻に開札まで到達できたが、当初表示の乗り換え時間2分はやはりはやいと感じた。

待ち時間の予測と実際の差異

質問3:電車や路線バス等の公共交通とオンデマンドバスを乗り換えた際に、乗換案内アプリが表示していた時間とどの程度のズレがありましたか？

実証	差異(分)									
	0~1	1~2	2~3	3~4	4~5	5~6	6~7	7~8	8~9	9~
全実証 (N=86)	38	12	8	15	4	4	2	2	0	1

5段階評価の結果

質問1:オンデマンドバスを含む適切な乗換経路を提案したと思いますか？

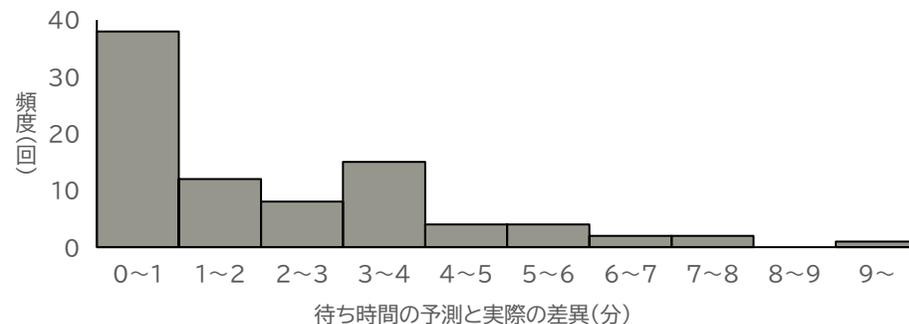
実証	回答数				
	非常にそう思う	そう思う	どちらでもない	あまり思わない	全く思わない
Checkpoint型 (N=50)	0	33	4	13	0
Zone型(N=36)	5	31	0	0	0

質問2:乗換待ち時間についての満足度は？

実証	回答数				
	非常にそう思う	そう思う	どちらでもない	あまり思わない	全く思わない
Checkpoint型 (N=50)	13	20	13	3	1
Zone型(N=36)	13	19	1	2	1

追加質問1:GTFS-Flexに加えGTFS-Ondemandも利用した方が利便性が良くなったと思いますか？

実証	回答数				
	非常にそう思う	そう思う	どちらでもない	あまり思わない	全く思わない
Checkpoint型 (N=14)	1	4	9	0	0
Zone型(N=15)	2	4	9	0	0



・その他、アンケートの自由回答の全回答は以下のとおりである。

質問4:乗換案内アプリが提案した経路について、ご意見やご感想をお書きください。

年齢	性別	回答
20代	女性	乗り合いが生じた場合予定の停留所ではなく近くの停留所にされるが、時間的に間に合わない可能性があると思う。また、予定では降車場所は目的地の近くのはずだったが、遠くの停留所にされそこから地下鉄やバスを使うようルート案内された。
20代	男性	ルート自体は良いが乗り換えが、間に合うか怪しいものがあった。もっと多くの候補から選ぶことに慣れているため、他にいいルートがあるんじゃないかと考える。
20代	男性	徒歩で行ったほうが絶対に近い経路を出されることもあれば、目の前の公共交通に乗れば良いのに歩かされる経路が出ることもある。終始信用が出来ない経路でした。
20代	男性	現実的でない経路が案内されたことがあった。
30代	女性	札幌は積雪時に特に歩くことが億劫なため、オンデマンドバスを含めた経路提案は非常によかったです。
30代	女性	既存のバスのデータが正しくない、駅とバスとデマンド交通と、あり得ない組み合わせでの提案があった
30代	男性	遅延が発生する場合や早着の可能性がある場合はプッシュ通知などがあると良い
30代	男性	明らかに遠回りする経路が提案されていた
40代	女性	次発時間の記載があるとありがたいです。
40代	女性	特にバスを含む乗り換えについて、乗り換えが多く手間になる経路が表示され戸惑った
40代	女性	自分では選ばないような、遠回りになるルートが提案された。旅行先など土地勘のない場所なら、表示されたとおりに移動していたと思う。
40代	男性	たまたま乗り換え回数が多くかなり遠回りな経路の時があり、徒歩やタクシーと迷うケースがでる気がした
40代	男性	非現実的な検索結果もあったものの、一定程度は納得した。一方で乗合によって当初の到着予定時刻を超過する事象が発生していたことから、実運用時には乗換が出来ないといったリスクも発生しうると感じた
50代	女性	土地勘がなく、納得いく提案かの判断はできませんでしたが、特にストレスに感じることもありませんでした。
50代	男性	乗車後の経路は、もっと地図に詳細が出るとよい。
50代	男性	歩く時間が多いので、こういった提案があると助かります。
50代	男性	活用されているデータやデマンド・人の位置により最適ルートが提案されなかったケースがあった
50代	男性	日常想定されないようなルートが結果としてでることがあった
60代	男性	想定しない場所がでてくる場合もあった。
60代以上	男性	3つ目に出てくるルートの到着時間が30~1時間程度遅いのは非現実的に思えた(検索時間による)アプリそのものは使いやすいと思う
60代以上	男性	車両の乗車時間が長めに思った(17分と出たが、3分程度で到着)

質問5:乗換待ち時間について、ご意見やご感想をお書きください。

年齢	性別	回答
20代	女性	降車予定がサッポロファクトリーだったが、札幌駅に変更されてしまった。また、早くは着くが乗り換えが一番多いルートが最初に出てきていた。
20代	男性	検索結果よりも、予定降車時刻は遅れていた。乗り換えは間に合わない。
20代	男性	早く車両が付いたので、そのまま乗車した
30代	女性	時間とスポットが限定される点が時代遅れ。
40代	女性	特に問題なくスムーズに移動できました
40代	女性	次のバスの乗り換えが6分と経路検索で表示されているところ、残り4分での到着だったので場所を把握していない人だと間に合わないと感じました
40代	女性	特に遅れもなく、乗車中運転手さんも和やかな雰囲気乗車体験としては良かったです
40代	女性	特に遅れなどなく順調に行ったので体験としては良かったです
40代	女性	初めて他の人が先に乗車しているところに乗ったが、FCコースターでバスの感覚があり、違和感なく乗れた。ただ降りるとき、スポットに着いたのか、何という名前のスポットなのか案内がなかったので、自分が降りると咄嗟に認識できず、他の人もいたので気まずい思いをした。慣れている場所であれば案内がなくても問題ないと思うが、特に乗り合いあったときはバスのようにスポット名案内してもらえると降りやすいと感じた。
40代	女性	予定の降車場所とは異なりましたが、体験としては特に問題なかったです。今回初めて他の人が先に降りることがあり、一瞬自分じゃないよね、は不安になりました
40代	女性	乗車体験としては問題なかった。バス停に行きたいのに、デマンドと電車であと歩き、という検索結果には違和感があった
40代	女性	検索した時、13:05で指定して13:14乗車が出てきたので、一瞬自分が指定時間を間違えたのかと思った。停まったとき外からは人がいるか分からないので、降りる人がいると気付いてちょっとびっくりした。乗ったときに行き先確認があり、また降車時も駐車場の列で止められない可能性を運転手さんから説明してくれたので安心でした。
50代	男性	乗り換え時間がギリギリだった
50代	男性	もし時間通りに運行しなかった場合、納得感がないかもしれない。
50代	男性	予約検索画面と完了画面で到着時刻がことなる。検索時に予定した市電にのれなかった
60代	男性	車両の到着時間が表示された時間より早かったため適切な時刻に開札まで到達できたが、当初表示の乗り換え時間2分はかなりはやいと感じた

追加質問2:GTFS-Flexに加えGTFS-Ondemandも利用したことによる利便性乗降ついて、ご意見やご感想をお書きください。

年齢	性別	回答
20代	男性	FLEXのみでは乗り継ぎに絶対間に合わない経路が(検索画面と異なり)予約された。
20代	男性	UIとしては大きく変わらないので、あまり利便性の面では変化を感じることは無かった。時刻の目安がOn-demandモードのほうがイメージできたことは良い点。
20代	男性	Flex-ondemandモードではデマンド交通の実態に沿った所要時間が表示されたため。
20代	男性	自分の乗りたい時間に乗れるため
30代	男性	ある程度時間信頼性が上がったため
30代	男性	提案されるデマンドの時間の確実性が高まるため
40代	女性	推定発着時間を考慮した場合と考慮していない場合の差が検索結果から読み取れませんでした。
40代	女性	今回の検証の中ではそんなに大きく違いは感じなかった
40代	男性	違いが分からなかった
40代	男性	いまいち違いがわからなかった
60代	男性	違いをあまり意識することが無かった

Webアプリとして達成すべき性能(応答時間やシステム稼働率等)をクリアし、類似のシステムを開発する際に参照可能な技術ナレッジを得ることができた

結果のまとめ

検証仮説

国際標準であるGTFS-Flex及びOndemandを活用し、国内で初めて実用に堪え得るマルチモーダル乗換検索システムを開発することで、その実装方法に関する技術ナレッジを社会に還元できる。

検証結果

本プロジェクトで設計・開発したマルチモーダル乗換検索アルゴリズムが、想定どおりに動作することを実装時の机上検証及び単体テスト(正常系テスト 20/20ケース、境界値テスト 20/20ケース)で確認し、実際の実証ケースでも問題なく動作することを検証した。

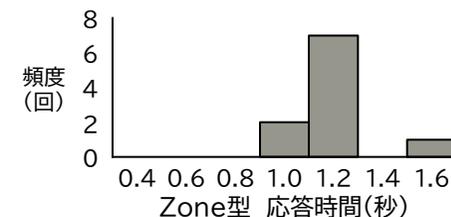
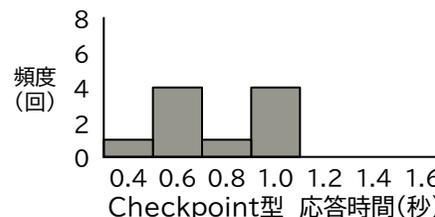
また、ランディングページのメインコンテンツが表示されるまでの時間は、実証実験期間内での実測で平均0.86秒、90パーセンタイル値で1.04秒であり、Webコンテンツとして十分な応答速度を発揮している。同様にマルチモーダル乗換検索結果画面の応答速度は平均2.22秒、90パーセンタイル値で2.54秒であり、グラフ探索やリアルタイム配車計算API呼出しといった複雑な処理を実行している画面にもかかわらず、Webコンテンツからの離脱を防ぐためのいわゆる「3秒ルール」以内で応答を返すことができている。

最後にシステム全体の稼働率であるが、実証実験の全期間を通じて障害によるシステム停止は発生せず、稼働率は100%となった。

得られた示唆

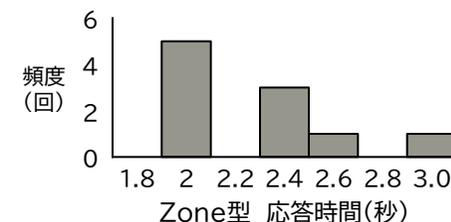
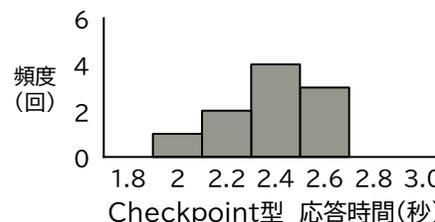
GTFS-Flex・Ondemandの実装事例及び技術ナレッジを創出

本プロジェクトにて設計・開発し、実証実験を実施したシステムでは、商用サービスである駅探乗換案内とSAVSを基に、GTFS-Flex及びGTFS-Ondemandを活用したマルチモーダル乗換検索アルゴリズムとそれを活用したUI/UXを、高い応答性能と安定した動作で実現した。これにより、一般的なWebアプリに慣れている利用者が離脱せずにご利用すると言われている3秒以下の画面応答速度をクリアし、実証期間を通じてシステム停止を全く発生させず安定したシステム運用を達成できた。これらの結果から、実運用に耐えうるシステムを開発できたと考えられ、今後類似のシステムを開発するエンジニアや企業・団体等にとって、実装事例として参照可能な有用な技術ナレッジが得ることができたと考えられる。



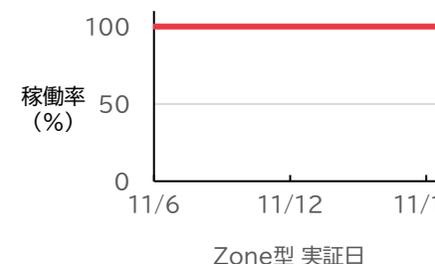
	Checkpoint型	Zone型	全体
90パーセンタイル(秒)	0.96	1.10	1.04
平均(秒)	0.65	1.06	0.86

ランディングページの応答時間計測結果



	Checkpoint型	Zone型	全体
90パーセンタイル(秒)	2.47	2.57	2.54
平均(秒)	2.28	2.16	2.22

乗換検索結果の応答時間計測結果



稼働率計測結果



検証方法

KPI詳細

KPI	定義	目標設定根拠
条件に合致するマルチモーダルな乗換経路が検索可能 :テストケースごとに設定された件数が検索される	テストケースごとの乗換結果探索数	・乗換パターンを網羅した正常系テストの件数 ・検索日時と乗降スポットの位置の境界値テストの件数

KPIの計測方法

- ・ 駅探乗換案内の開発者1名が、駅探乗換案内を改修して実装したマルチモーダル乗換検索機能とテスト用SAVS APIを接続し、アルゴリズム単体での動作確認を行った。
- ・ 具体的な検証フローは以下の通り
 - 要件定義に従い、マルチモーダル乗換検索機能を実装する。
 - SAVS APIの仕様に従い、事前に指定した結果を返すテスト用APIを実装する。
 - マルチモーダル乗換検索の検索条件とSAVS APIの配車計算結果の組み合わせにより、乗換パターンを定義する。
 - マルチモーダル乗換検索機能とテスト用SAVS APIを接続し、乗換パターンごとに想定した乗換検索結果が得られることを確認する。ただしGTFS-Flexのみを用いる場合は、テスト用SAVS APIと接続しない。
 - サービス稼働時間外やサービスエリア外での乗換経路検索など、境界となる条件の前後で乗換経路を検索し、正しく動作することを確認する。

#	検索条件	概要
①	出発地(地図から選択)	地図から選択した出発地の緯度・経度
②	出発地(乗降スポットから選択)	事前に定められた乗降スポットの緯度・経度
③	出発地(駅・バス停から選択)	定時定路線交通の駅・バス停
④	目的地(地図から選択)	地図から選択した目的地の緯度・経度
⑤	目的地(乗降スポットから選択)	事前に定められた乗降スポットの緯度・経度
⑥	目的地(駅・バス停から選択)	定時定路線交通の駅・バス停
⑦	出発希望日時	指定する出発希望日時
⑧	到着希望日時	指定する到着希望日時

#	SAVS APIの配車計算結果
①	配車車両のID、推定乗車日時、推定降車日時、最遅の推定降車日時
②	配車可能な車両が存在しない

結果の詳細

マルチモーダル乗換検索アルゴリズムの動作検証結果

- ・ GTFS-Flex及びGTFS-Ondemandを活用する乗換パターン4種とGTFS-Flexのみ利用する乗換パターン2種に対して検索条件とSAVS APIの結果を網羅的に組み合わせ、正常系テストを行った。
- ・ 潜在的なバグを洗い出すために、日付や位置情報の境界値の前後をチェックするテストケースも洗い出し検証を行った。

乗換パターン1

定時定路線交通→オンデマンドバス、出発時間指定【GTFS-Flex+GTFS-Ondemand】

#	検索条件								API結果		想定する結果	合否
	①	②	③	④	⑤	⑥	⑦	⑧	①	②		
1			○		○		○		○	○	Checkpoint型 検索される乗換経路1件	合格
2			○	○			○		○	○	Point-to-zone型 検索される乗換経路1件	合格
3			○		○		○		○	○	Checkpoint型 検索される乗換経路0件	合格
4			○	○			○		○	○	Point-to-zone型 検索される乗換経路0件	合格

乗換パターン2

オンデマンドバス→定時定路線交通、出発時間指定【GTFS-Flex+GTFS-Ondemand】

#	検索条件								API結果		想定する結果	合否
	①	②	③	④	⑤	⑥	⑦	⑧	①	②		
5		○				○	○		○	○	Checkpoint型 検索される乗換経路2件	合格
6	○					○	○		○	○	Zone-to-point型 検索される乗換経路2件	合格
7		○				○	○		○	○	Checkpoint型 検索される乗換経路0件	合格
8	○					○	○		○	○	Zone-to-point型 検索される乗換経路0件	合格



結果の詳細

乗換パターン3

定時定路線交通→オンデマンドバス、到着時間指定【GTFS-Flex+GTFS-Ondemand】

#	検索条件								API結果		想定する結果	合否
	①	②	③	④	⑤	⑥	⑦	⑧	①	②		
9			○		○			○	○		Checkpoint型 検索される乗換経路1件	合格
10			○	○				○	○		Point-to-zone型 検索される乗換経路1件	合格
11			○		○			○		○	Checkpoint型 検索される乗換経路0件	合格
12			○	○				○		○	Point-to-zone型 検索される乗換経路0件	合格

乗換パターン4

オンデマンドバス→定時定路線交通、到着時間指定【GTFS-Flex+GTFS-Ondemand】

#	検索条件								API結果		想定する結果	合否
	①	②	③	④	⑤	⑥	⑦	⑧	①	②		
13		○				○		○	○		Checkpoint型 検索される乗換経路1件	合格
14	○					○		○	○		Zone-to-point型 検索される乗換経路1件	合格
15		○				○		○		○	Checkpoint型 検索される乗換経路0件	合格
16	○					○		○		○	Zone-to-point型 検索される乗換経路0件	合格

乗換パターン1'

定時定路線交通→オンデマンドバス、出発時間指定【GTFS-Flexのみ】

#	検索条件								API結果		想定する結果	合否
	①	②	③	④	⑤	⑥	⑦	⑧	①	②		
1			○		○			○			Checkpoint型 検索される乗換経路1件	合格
2			○	○				○			Point-to-zone型 検索される乗換経路1件	合格

結果の詳細

乗換パターン4'

オンデマンドバス→定時定路線交通、到着時間指定【GTFS-Flexのみ】

#	検索条件								API結果		想定する結果	合否
	①	②	③	④	⑤	⑥	⑦	⑧	①	②		
3		○						○	○	○	Checkpoint型 検索される乗換経路1件	合格
4	○							○		○	Zone-to-point型 検索される乗換経路1件	合格

※ GTFS-Flexのみを用いる場合、オンデマンドバスのリアルタイム配車計算を行わないため、オンデマンドバスの降車時間が確定しない。そのためオンデマンドバスの推定降車日時を基に定時定路線交通の乗換経路を検索する乗換パターン2及び乗換パターン3は、GTFS-Flexのみの場合はアルゴリズムの動作検証対象外。

※ GTFS-Flexのみを用いる場合、オンデマンドバスのリアルタイム配車計算をしないため、テスト用SAVS APIが配車可能な車両がない結果を返す②はアルゴリズム動作検証対象外。

境界値テスト

#	検索条件	想定する結果	合否
1	オンデマンドバス利用開始期間の開始日前日	検索される乗換経路無し	合格
2	オンデマンドバス利用開始期間の開始日当日	検索される乗換経路有り	合格
3	オンデマンドバス利用開始期間の終了日当日	検索される乗換経路有り	合格
4	オンデマンドバス利用開始期間の終了日翌日	検索される乗換経路無し	合格
5	オンデマンドバス利用不可曜日(土曜)	検索される乗換経路無し	合格
6	オンデマンドバス利用不可曜日(日曜)	検索される乗換経路無し	合格
7	オンデマンドバスサービス開始時刻の1分前	検索される乗換経路無し	合格
8	オンデマンドバスサービス開始時刻ちょうど	検索される乗換経路有り	合格
9	オンデマンドバスサービス終了時刻ちょうど	検索される乗換経路有り	合格
10	オンデマンドバスサービス終了時刻の1分後	検索される乗換経路無し	合格
11	出発する乗降スポットstop_idがstops.txtに未定義	検索される乗換経路無し	合格
12	出発する乗降スポットstop_idがstops.txtに定義済	検索される乗換経路有り	合格
13	到着する乗降スポットstop_idがstops.txtに未定義	検索される乗換経路無し	合格
14	到着する乗降スポットstop_idがstops.txtに定義済	検索される乗換経路有り	合格
15	出発する乗降スポットがlocations.geojsonエリア外	検索される乗換経路無し	合格
16	出発する乗降スポットがlocations.geojson境界上	検索される乗換経路有り	合格
17	出発する乗降スポットがlocations.geojsonエリア内	検索される乗換経路有り	合格
18	到着する乗降スポットがlocations.geojsonエリア外	検索される乗換経路無し	合格
19	到着する乗降スポットがlocations.geojson境界上	検索される乗換経路有り	合格
20	到着する乗降スポットがlocations.geojsonエリア内	検索される乗換経路有り	合格



検証方法

KPI詳細

KPI	定義	目標設定根拠
ランディングページ応答時間:3秒以下(90パーセンタイル値)	ランディングページのLCPの90パーセンタイル値	一般的に3秒を越えるとページからの離脱率が上がるため

LCP(Largest Contentful Paint): 画面の主要なコンポーネントのレンダリングが完了した時間

KPIの計測方法

- 開発統括担当1名が、Checkpoint型及びZone型の実証実験期間中に、実証で用いているシステムの画面を用いてランディングページの応答時間を実測した。
- 具体的な検証フローは以下の通り
 - デスクトップPCを携帯電話網(4G LTE)へテザリング接続する。
 - Google Chromeを起動し、Developer ToolsのLighthouseを開く。
 - ランディングページをロードし、LCPを計測する(キャッシュの効果を得るために、1回目の計測結果は破棄)。
 - Checkpoint型及びZone型の実証実験期間中に計測を10回繰り返し、ランディングページの応答時間の90パーセンタイル値を算出。

結果の詳細

ランディングページ応答時間の計測結果

Checkpoint型		Zone型	
計測	応答時間(秒)	計測	応答時間(秒)
1	0.78	1	1.02
2	0.40	2	1.02
3	1.00	3	1.00
4	0.96	4	1.46
5	0.44	5	1.01
6	0.42	6	1.02
7	0.82	7	1.04
8	0.42	8	1.06
9	0.87	9	1.02
10	0.42	10	0.97

検証方法

KPI詳細

KPI	定義	目標設定根拠
乗換検索結果ページ応答時間:7秒以下(90パーセンタイル値)	乗換探索結果ページの応答時間の90パーセンタイル値	過去事例からの推定

KPIの計測方法

- 開発統括担当1名が、Checkpoint型及びZone型の実証実験期間中に、実証で用いているシステムの画面を用いて乗換検索結果ページの応答時間を実測した。
- 具体的な検証フローは以下の通り
 - デスクトップPCを携帯電話網(4G LTE)へテザリング接続する。
 - Google Chromeを起動し、Developer ToolsのNetworkを開く。
 - マルチモーダルな乗換検索を行って乗換検索結果ページまで遷移し、乗換検索結果データがダウンロードされるまでの通信時間を計測する(キャッシュの効果を得るために、1回目の計測結果は破棄)。
 - ※ 駅探乗換案内アプリはSPAのため、データ取得と同時に画面が書き換わる
 - Checkpoint型及びZone型の実証実験期間中に計測を10回繰り返し、乗換検索結果ページの応答時間の90パーセンタイル値を算出。

結果の詳細

乗換検索結果ページ応答時間の計測結果

Checkpoint型		Zone型	
計測	応答時間(秒)	計測	応答時間(秒)
1	2.55	1	1.91
2	2.47	2	2.54
3	2.07	3	1.90
4	2.30	4	1.86
5	2.40	5	2.24
6	2.33	6	1.90
7	2.28	7	2.22
8	1.97	8	1.87
9	2.06	9	2.87
10	2.35	10	2.27



検証方法

KPI詳細

KPI	定義	目標設定根拠
実証実験期間中の稼働率(システム計画停止は含まず):95%/月	ログから算出するシステム障害開始時間と終了時間の差分	実証用のため、月間1日程度のトラブル対応を見込む

KPIの計測方法

- 駅探乗換案内にログインしてマルチモーダルな乗換検索を自動的に実行するスクリプトを作成し、実証期間中定期的(5分に1回)に動作させ、システムが正常に動作していることを確認した。
- 具体的な検証フローは以下の通り
 - Pythonを用いて、駅探乗換案内にログインして検索条件を入力し、マルチモーダルな乗換検索を自動的に実行してその検索結果のHTTPレスポンス成否をログに記録するスクリプトを作成する。
 - Checkpoint型及びZone型の実証実験期間中、AWS Lambdaを用いて5分に1回定期的なスクリプトを実行し続け、その結果をAWS CloudWatch Logsに記録する。
 - 実証実験期間終了後、ログを集計してシステム全体の稼働率を算出する。

結果の詳細

システム稼働状況の計測結果

Checkpoint型

計測日	成功回数	失敗回数	稼働率
11月6日	288	0	100%
11月7日	288	0	100%
11月8日	288	0	100%
11月9日	288	0	100%
11月10日	288	0	100%
11月11日	288	0	100%
11月12日	288	0	100%
11月13日	288	0	100%
11月14日	288	0	100%
11月15日	288	0	100%
11月16日	288	0	100%
11月17日	288	0	100%
11月18日	288	0	100%
11月19日	288	0	100%

Zone型

計測日	成功回数	失敗回数	稼働率
10月10日	288	0	100%
10月11日	288	0	100%
10月12日	288	0	100%
10月13日	288	0	100%
10月14日	288	0	100%
10月15日	288	0	100%
10月16日	288	0	100%
10月17日	288	0	100%
10月18日	288	0	100%
10月19日	288	0	100%
10月20日	288	0	100%
10月21日	288	0	100%
10月22日	288	0	100%



第5章 まとめ

GTFS-Flex及びGTFS-Ondemandを活用し、乗換検索のUI/UXから定時定路線交通とオンデマンドバスとにまたがったマルチモーダルな乗換検索を実施でき、検索結果からシームレスにリアルタイムでオンデマンドバスの配車予約まで実施できるシステムを開発し、実証実験を行った。その結果、オンデマンドバスの発見性や利用性が向上し、社会実装への期待も高いことが判明した。この成果を広く一般に公開するとともに、UI/UXの進化と実績を重ね、ベストプラクティスとして共有し採用実績を増やしていくことで、地域住民のみならず旅行者からも利用される「公共交通の使いやすさ」の新しいスタンダードが実現できると期待している。

GTFS-Flex/GTFS-Ondemandを活用することでオンデマンドバス利便性が向上し、利用機会の増大、持続可能な公共交通の実現へ寄与できることが明らかとなった

得られた成果

本プロジェクトを通じて、定時定路線交通とオンデマンドバスとにまたがるマルチモーダルな乗換検索と、乗換検索結果からシームレスにオンデマンドバスの配車が予約できるシステムが、オンデマンドバスの利用増大に寄与できることが分かった。これによりオンデマンドバス事業を営む交通事業者の収益性の改善も期待でき、地方公共団体の助成に頼り切らない持続可能な地域の足としての展開も視野に入る。

GTFS-FlexにGTFS-Ondemand活用した乗換検索・配車予約システムの実現

GTFS-FlexにGTFS-Ondemand拡張を加えた15個の静的なファイルにより、札幌市創成イーストエリアでの実証実験の要件に合致したオンデマンドバス運行情報を過不足無く定義できた。これらの静的なファイルやGTFS-Ondemandのリアルタイムデータを活用し、乗換案内システムとオンデマンドバス配車予約システムを容易に連携でき、オンデマンドバス運行に関する多彩な情報を利用者に提供できた。

動的メッセージを利用したマルチモーダル乗換検索アルゴリズムの実現

上記オンデマンドバス対応GTFSの静的なファイルから得たオンデマンドバスの静的な運行情報に、GTFS-Ondemandが拡張した動的なメッセージ形式を活用したリアルタイムなオンデマンドバスの配車情報を加え、定時定路線交通のグラフ探索アルゴリズムと連携したマルチモーダル乗換検索アルゴリズムを実現し、実証実験を通じて安定して動作することを確認した。ただし、社会実装に向けては、乗換経路の精緻化や乗換待ち時間予測の最適化など、実地検証による調整が必要な余地も残る。

GTFS-Flex/GTFS-Ondemand活用によるオンデマンドバスの利便性向上

マルチモーダルな乗換案内とシームレスでリアルタイムな配車予約による実証実験を札幌市で実施した結果、適切なオンデマンドバス経路確認と、9割を超える被験者から非常に高い評価を得た。本プロジェクトで体験したサービスが実現した際は利用したいと回答した被験者も9割を超え、実装への高い期待が確認できた。

上記の本プロジェクト成果に加え、GTFS-Flex及びGTFS-Ondemandのデータ項目が利用者への情報提供や乗換検索に有用であった一方、オンデマンドバスの予約APIが現時点ではGTFS系エコシステムに含まれておらず個別開発が必要であったこと等をGTFSの国際標準化団体であるMobility Dataにフィードバックした。Mobility Dataからは、乗換検索を提供するCP、配車予約サービス事業者及び運行事業者という異なる関係者を捉え、検索から予約、決済に至る一連のプロセスをひとつの実装で整理した貴重な成果であり、今後参考にしたいと評価を得ている。

得られたナレッジのまとめ

オンデマンドバスに対応したGTFSデータの作成

- GTFS-FlexにGTFS-Ondemand拡張を加えた15個の静的なデータファイルにより、実証実験の要件に即したオンデマンドバスの運行情報を定義できた。
- GTFSに関する基礎知識を持つITエンジニアであれば、これらの静的なデータファイルの作成とチェックを3人日程度で実施できることが分かった。

オンデマンドバス対応したGTFSを利活用したシステムの開発

- GTFS-Ondemandが拡張した動的なメッセージ形式(WaitTimeUpdate及びFareUpdate)を用いることで、SAVSから駅探乗換案内へオンデマンドバスの待ち時間情報や運賃情報を提供できることが分かった。
- 実証実験期間中のシステム障害停止は無く、稼働率100%を達成した。
- ランディングページのみならず、複雑なアルゴリズムを内包するマルチモーダル乗換検索結果画面を含め、応答時間は3秒以下と高いパフォーマンスを発揮した。

マルチモーダル乗換検索アルゴリズムの開発

- グラフ探索アルゴリズムによる定時定路線交通の経路探索にオンデマンドバスのリアルタイム所要時間推定をつなぎ込むことで、マルチモーダルな乗換検索アルゴリズムが実現できた。

オンデマンドバスの利用機会増大につながるUI/UXの開発

- 駅探乗換案内のUIを改修することで、広く一般に使われている乗換検索のUI上で定時定路線交通とオンデマンドバスをまたがるマルチモーダル乗換検索を行い、そのままシームレスにオンデマンドバスの配車を予約できるUI/UXが実現できた。

本プロジェクトの成果物

- GTFS-Flex及びGTFS-Ondemandの技術実証プロジェクトプロジェクトレポート
- https://www.mlit.go.jp/commmmons/projectreport/09_01/
- GTFS-Flex/Ondemand活用システム 技術検証レポート
- https://www.mlit.go.jp/commmmons/tech_report/009/
(付録)GTFS-Flex/Ondemand活用システム システム設計書
(付録)GTFS-Ondemand仕様案(日本語訳)

将来展望

関係者が一体となり、持続可能な地域の足としてのオンデマンドバスをさまざまな利用者がより簡単に手軽に利用できる社会の実現を目指す

UI/UXの高度化と技術的要件の深化

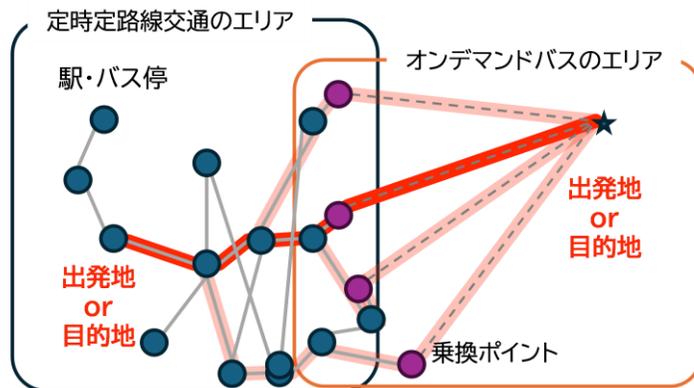
実証実験で得た知見を基に、社会実装に耐えうる水準までUI/UXを洗練させる。具体的には、より自然な乗換経路を提示するアルゴリズムの最適化や、遅延情報のリアルタイムなプッシュ通知など、ユーザーの不安を解消する機能を実装する。また、検索から予約まで定みなく繋がる「情報断絶がないUX」を標準化し、ノウハウを業界全体へ共有することで、オンデマンドバスのデファクトスタンダード構築を目指す。

多様なユースケースへの対応と汎用性の確立

特定の季節や地域に限定されない、汎用性の高いシステムの構築を推進する。厳冬期の積雪による渋滞環境や、「交通空白」のある郊外など、異なる条件下での試行を継続する。地域特性に応じた多様な成功事例(ユースケース)を積み上げ、いかなる環境下でも機能する「持続可能な移動インフラ」としての信頼性を担保する。

持続可能な地域公共交通エコシステムの構築

システム開発者や配車事業者等の技術提供側のみならず、地域の交通事情を熟知したコンサルタント、地場の交通事業者、自治体が一体となった推進体制を組織する。開発・運用・保守が地域内で完結し、循環する「自律的なエコシステム」を構築することで、補助金に過度に依存しない、持続可能な地域公共交通の社会実装を実現する。



定時定路線交通とオンデマンドバスの統合的な検索とオンデマンドバスのリアルタイム配車予約を実現



オンデマンドバスの運行風景



定時定路線バスとオンデマンドバスの運行風景



**GTFS-Flex/Ondemand活用システム 技術検証レポート
Ver1.0**

発行日: 2026年3月

委託者: 国土交通省 総合政策局
情報政策課、モビリティサービス推進課

受託者: TIS株式会社