

動的バス情報フォーマット（GTFSリアルタイム）ガイドライン

本ガイドラインには、GTFSリアルタイムフォーマットに基づいてデータを提供および利用する際に推奨する方法について記載します。

本ガイドラインは、データを提供および利用する情報処理技術者を主な対象読者として作成しています。

1. 制定の経緯

バスの運行は道路混雑状況等により日常的に遅延が発生することが多いため、遅延情報や位置情報等の動的情報は利用者にとって極めて重要な情報です。

動的情報はバスロケーション機器により情報収集しますが、これまでは機器から発信・収集するデータについての基準やルールなどは何もありませんでした。このため、バス事業者とバスロケーションメーカー間の個別の交渉で取り扱うデータが決定され実装されてきています。その結果、バス事業者毎に取り扱うデータ項目等が多種多様となっているのが実状です。

一般的に動的情報は各バス事業者のHPや自社アプリケーションなどにより利用者に提供されています。このような自社内に限定した利用であれば特段の課題はないのですが、複数のバス事業者の運行情報を統合したデジタルサイネージを整備する場合に、一部の事業者が遅延情報などの必要なデータを収集しておらず統合表示が難しいなどの問題が生じています。

その他にもデータにばらつきが大きいと、関係者間でデータを連携・共有しようとした際に様々な不具合が生じることが想像されます。そこで、2018年度に開催した「バス情報の静的・動的データ活用検討会（座長：東京大学・伊藤昌毅助教）」での議論を踏まえ、『GTFSリアルタイム』を動的情報の標準的なフォーマットとして定め、「標準的なバス情報フォーマット」に追加しました。

バスロケーションシステムを新規導入あるいは機器更新する際には、これに基づいて整備されることを国として推奨します。

なお、「都市と地方の新しいモビリティサービス懇談会（座長：筑波大学・石田東生特命教授）」が2019年3月14日に公表した中間とりまとめでも、MaaS事業者や交通事業者間でデータ連携を進める上でデータ形式の標準化の有効性が記載されています。

2. GTFSリアルタイムについて

2.1. フォーマットの概要

GTFSリアルタイムとは公共交通のリアルタイム情報を格納するためのフォーマットです。GTFSリアルタイムデータは単独では機能せず、GTFS(-JP)データと併せて利用します。

2.2. 内容

動的データには下記の情報を含めることができます。

対象	要素	設定可能な主な情報
ルート最新情報	TripUpdate	遅延、発着時刻予測、通過
車両位置情報	VehiclePosition	車両の緯度・経度、接近情報、混雑度
運行情報	Alert	運行情報の概要、影響（運休、迂回等）、原因（天候、事故等）、URL

これら3つの要素ごとに別ファイルにしても、1つのファイルに格納しても構いません。

2.3. データ形式

GTFSリアルタイムはProtocol Buffersという、データ構造が規定されたバイナリ形式をベースとしています。

データ構造は「gtfs-realtime.proto」というテキストファイルにて定義されます。

2.4. ベースとするバージョン

- 2019年3月時点の「標準的なバス情報フォーマット」の動的データフォーマットは、GTFSリアルタイム v2.0をベースとします。
 - 日本語文書：Google Developersのサイト
 - <https://developers.google.com/transit/gtfs-realtime/reference/>
 - データ構造：gtfs-realtime.proto
 - <https://developers.google.com/transit/gtfs-realtime/gtfs-realtime.proto>
- GTFSリアルタイム v2.0以降のバージョンがリリースされた場合は、その利用を妨げるものではありません。

3. 設定方法

GTFSリアルタイムフォーマットにおいて推奨する値の設定方法、およびその利用方法について記載します。

3.1. バスロケーション情報として出力する項目

- 設定方法
 - TripUpdate（ルートの最新情報）とVehiclePosition（車両の現在位置）の両方を出力するようにしてください。
- 理由
 - 経路検索サービス等においてはTripUpdateに含まれる遅延や発着時刻予測等を必要とし、地図表示においてはVehiclePositionに含まれる車両の緯度・経度を必要とするなど、どちらの要素も必要性が高いためです。

3.2. 遅延時間(delay)

- 設定方法
 - 遅延時間（TripUpdate.delay, StopTimeEvent.delay）については、分単位などの丸め処理（例：143秒を120秒にする）や、一定時間未満の切り捨て（例：36秒を0秒にする）を行わないことが望ましいです。
- 理由
 - 表示の工夫はデータ利用側において行われるべき処理だからです。

- 利用側にて到着予測などのデータ分析を行う場合に、予め丸め処理がされていると精度が落ちるからです。
- 利用する際の留意点
 - リアルタイム情報ではあるもの、伝送遅延や予測誤差により1分程度の誤差が発生していることを想定し、乗り遅れを防ぐような表記をしてください。
 - 例：到着予測まで120秒未満となった場合に「まもなく到着」

3.3. 時刻の不確実性(uncertainty)

- 設定方法
 - 通過済の停留所に対する、時刻の不確実性（StopTimeEvent.uncertainty）の値はゼロにしてください。
 - 未通過の停留所に対する、時刻の不確実性（StopTimeEvent.uncertainty）の値は秒単位の正の値にしてください。
- 理由
 - 利用側にて通過済かどうかを判定しやすくするためです。

4. 配信方法

GTFSリアルタイムデータを配信および利用する際に推奨する方法について記載します。

4.1. データの更新間隔

- 配信方法
 - データ更新間隔は30秒以下としてください。これは最低限の既定であり、より短い更新間隔であることが望ましいです。
 - データ配信サイト等に、更新間隔を記すようにしてください。
- 理由
 - 更新間隔が長すぎると、到着予測などの誤差が大きくなり、乗り遅れ等が発生してしまうため。
 - 既存バスロケーションシステムの更新間隔の多くが30秒以下であり、最低限準拠可能な設定と考えられるため。
- 利用する際の留意点
 - フィードや測定の日付タイムスタンプ（TripUpdate.timestamp, VehiclePosition.timestamp, FeedHeader.timestamp）が、現在時刻よりも著しく古い場合は、異常値と考えられるため、除去するなどの工夫をしてください。
- 参考情報
 - バスロケーションに関する情報は、「車載器 → バスロケーションシステムサーバ → 情報提供サーバ → 利用者端末」というような経路で車両から利用者に伝送されます。この過程の中で、バッファリング、送受信待ち、データ処理、通信などの時間が積み重なります。
 - システム全体としての遅延時間を抑えるために、データ提供者、データ利用者双方において、無用なバッファリングや送受信待ちをしないよう留意してください。
 - 遅延を抑える必要がある場合には、プッシュ技術を用いるなどの工夫をしてください。

4.2. エンドユーザからのデータ取得方法

- 利用する際の留意点
 - GTFSリアルタイムデータを用いた一般向け情報提供サービスを開発する際は、エンドユーザからデータ配信元に直接データを取りに行くのではなく、間に情報提供サービスのサーバを介すようにしてください。

い。

- 理由
 - データ配信元の負荷を下げるためです。
- 参考情報
 - 2019年3月現在、オープンデータとして配信されているGTFSリアルタイムデータは、いずれも公開サーバ上に配置されております。
 - 過度なアクセスをしないよう、データ配信サイトに注意事項を記している例があります。
 - 例) 15秒おきに更新されます。過度なアクセスは行わないようにしてください。

5. 補足事項

5.1. 車両と時刻表の紐付けについて

- ルート最新情報、車両位置情報については、時刻表との紐付け（trip_id）が事実上必須となっています。
- このため、下記のような場合にデータの作成が困難です。
 - 災害時
 - 通常時のダイヤが機能しておらず、時刻表と紐付けられない。
 - 臨時便
 - 時刻表にない臨時バスのため、時刻表と紐付けられない。
 - 簡易なバスロケーションシステム
 - 系統、通過停留所、座標などしか保持しておらず、仕業や時刻表と紐付いていない簡易なバスロケーションシステム
- このような状況で出力されるデータの取り扱い方法については、2019年3月時点では今後の検討課題です。