オープンデータの階層化によるアプリケーション表示速度の改善

Improving Application Display Speed through Hierarchical Structuring of Open Data

望月　和馬
指導教員　細野　繁

東京工科大学　コンピュータサイエンス学部　コンピュータサイエンス学科
サービスシステムデザイン研究室

日オープンデータを階層化することで、地図アプリケーションにおける表示速度の改善を目指すものだ。
必要な情報のみをリアルタイムで表示し、詳細データは操作に応じて動的に読み込む手法を提案する。
これにより、大量のデータ処理負荷を分散させてより迅速な情報提供を実現する。

キーワード：オープンデータ，地図アプリケーション，階層化データ

1．　序論

現代の都市開発やサービス提供において，オープンデータの活用はますます重要となっている．オープンデータとは国や地方公共団体が公開しているデータであり，二次利用が可能，著作権が存在しない，機械判読が可能といった特徴がある．そのためデータを生かしたサービスを作ることができる反面，膨大な量のデータ更新の必要性からその処理や管理には多くの課題が存在する．

2．研究目的

本研究の目的は，膨大な量のオープンデータと地図データを活用したサービスにおいて，リアルタイム性を向上させるシステムを開発することだ．特に，交通管理や観光客向けの情報提供サービスを例に，IoT デバイスから収集されるデータを加えることでサービスの質をより高めた状態でのリアルタイムな情報提供を実現することを目指す．

3．提案手法

本研究では，オープンデータが抱える問題の一つであるリアルタイム性の確保を目指し，データ処理速度の向上を図る手法を提案する．具体的には，都市 OS を用いたデータ要求の管理において，データの階層化を実施し，効率的なデータ管理を行う．これにより，一度に処理するデータ量の削減と迅速なデータ提供を実現することを目指す．そのシステム構築図を図 1 に示す．FIWARE は，都市のインフラやサービスを統合・管理するためのソフトウェアプラットフォームだ．本研究では，都市 OS を活用して効率的にデータを取得・管理し，そこから迅速な情報提供を行ってアプリケーションの描画速度を向上させるシステムを構築する．

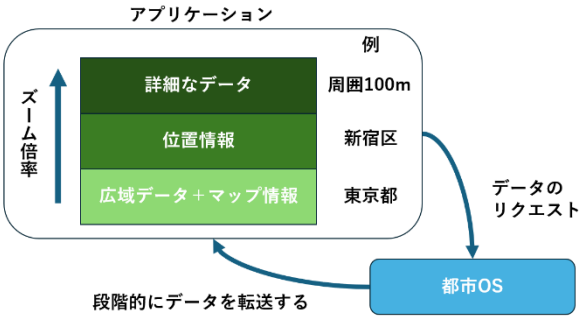

図 1：表示速度を向上させるための取り組み

3．1　階層化データの使用

データを階層化し，ユーザーのズームレベルに応じて適切な詳細度のデータを提供する手法を提案する．初期のズームアウト時では広域データのみを読み込むことで処理するデータ量を減らし，最初の地図を描画するスピードを早める．その後ユーザーがズームインしていくにつれ，施設の位置情報や詳細なデータを順次読み込んで表示する．例えば，広域データでは大まかな範囲内の施設の

数を表示し，ユーザーのズーム操作に応じて位置情報や営業時間などのデータを読み込んで表示する．この方法により，広域ビューではデータ量を削減し，詳細ビューでは必要な情報を提供することが可能だ．これにより，全体的なデータ処理負荷を分散させ，ユーザーに対する迅速な応答を実現する．

4．検証

提案した手法を検証するための実行環境として，3 つのシステムを使用している．オープンデータおよび IoT データの取得･保存を行う FIWARE，FIWARE から取得したデータを処理するアプリケーションとして機能する jupyterLab，地図データや位置情報サービスを提供するプラットフォームである HERE API の 3 つだ．このシステムを連携させることで，FIWARE から送られてきたデータの位置情報を読み取り，jupyterlab を介して HEREAPI の地図上にリアルタイムでマーカーを表示できる．このプロセスの処理時間を測定し，提案手法の有効性を評価する．以下に，実際に FIWARE から送られてきたデータを基に HERE API の地図上に位置情報マーカーを追加した例を示す．

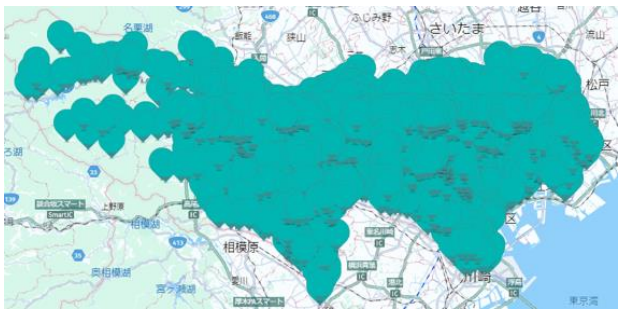

図 2：実際の動作画面

5．評価

データを受け取ってから処理を開始して地図上に可視化できるまでの時間を計測し，どれぐらい短縮できたか比較を行う．現在のシステムと改善案を実装した後の時間を計測することで，何もしていない処理方法に比べて 10%の速度が上昇した場合に，本研究で提案した手法でデータ処理の時間を早めることができたと判断する。

6．これまでの調査

これまでの調査として，警視庁から発表されている交通事故の位置情報約 5000 件ほどのデータをマップに表示させた際のそれぞれの段階の処理間を表 1 に示す．具体的には，FIWARE から jupyterlab にデータを転送する時間，転送されたデータの位置情報を参考に地図上にマーカーを追加して表示するまでの時間，その後地図を描画する時間をまとめた．

表 1：それぞれの段階の所要時間

項目	測定値	備考
データ転送	2.37 秒	FIWARE から転送
マーカー追加	151.70 秒	リストから地図へ登録
地図表示	3.57 秒	完成図表示
総合処理時間	157.64 秒	性能によるズレあり

現段階の調査において，HTTP リクエストを用いた FIWARE からアプリケーションへのデータ転送速度は非常に速いことが確認できた．しかし，実際に受け取った位置情報を地図上に表示する処理に多くの時間がかかっていることが判明している．

7．今後の計画

これまでの調査から，FIWARE からアプリケーションにデータを転送する速度は十分に速い一方で，受け取った位置情報を地図上に表示する処理に多くの時間がかかっていることがわかった．したがって，今後の計画としては，この表示処理時間を短縮するための手法を検討し，提案手法の有効性を検証する．具体的には，位置情報をリアルタイムで効率よく地図上に可視化できるような改善を加え，処理速度の向上を目指す．

参考文献

[1] Kishida, Keisuke. ”Python を使用して物流 DX のプロトタイピングをしてみる.” QIITA. https://qiita.com/kekishida/items/872a95079503ff36b544 (2024 年 7 月 31 日/覧)