

Title	空間情報活用のためのデータ統合基盤に関する研究
Author(s)	亀田, 晃佑
Citation	
Issue Date	2022-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17647
Rights	
Description	Supervisor: 丹 康雄, 先端科学技術研究科, 修士(情報科学)

修士論文

空間情報活用のためのデータ統合基盤に関する研究

亀田晃佑

主指導教員 丹 康雄 教授

北陸先端科学技術大学院大学
先端科学技術研究科
(情報科学)

令和4年3月

Abstract

Spatial information refers to the pairing of information on the location of objects in the real world with information on attributes other than location, and there are various types, from static to dynamic. By combining multiple types of data, spatial information that cannot be recognized with a single type of data can be captured and utilized in applications. For example, a navigation system is an application that utilizes multiple types of spatial information to search for a route to a destination by overlaying the location of a moving object obtained by position detection systems such as GNSS (Global Navigation Satellite System) and POI (Points of Interest) that represent destinations, in addition to "base maps" or "satellite images" of the Geospatial Information Authority of Japan.

When realizing such an application that utilizes multiple spatial information, the application developer needs to collect all the necessary spatial information and realize data processing and analysis on the application side to deal with differences in syntax and semantics among the data.

In this study, we propose a data integration infrastructure for the utilization of spatial information. This infrastructure has three major functions: data collection, data processing and analysis, and data provision. In the "data collection" function, data owned by various data providers are collected and registered in a spatial database on the infrastructure. The data collected in the infrastructure is provided via the "data provision" function in response to requests from users who develop applications. At this time, the user can set the "data processing and analysis" function as necessary, and can obtain output data from multiple data processing and analysis procedures.

There are two possible ways to use the data acquired by the user. The first is to use the acquired data in their own applications. The second, the acquired data can be processed as necessary and registered again in the data integration infrastructure. This makes it possible for other users who are considering using similar data to acquire the data without having to go through the data processing and analysis procedures that were required to create the data.

By acquiring data using the functions of the proposed data integration infrastructure, we can not only reduce the number of individual data collection and data processing/analysis procedures that had to be performed on the application side, but also reduce the resources required for data and processing/analysis. In addition, since this data integration infrastructure provides a single data source

and data processing and analysis functions commonly to multiple users, it can eliminate the need to duplicate data and data processing and analysis functions among applications.

In this study, we implemented an API for users to acquire data from the infrastructure, and verified its operation for two scenarios: "confirming the number of people waiting for a bus" and "confirming the airspace available for drone flights. As a result, we were able to reduce the data collection, data processing, and analysis procedures on the application side by 60% and the amount of data retained by the application side by 99.8% in the first scenario, and similarly, we were able to reduce the data collection, data processing, and analysis procedures by 90% and the amount of data by 75% in the second scenario.

概要

空間情報とは、実世界におけるモノの位置情報と、位置以外の属性の情報が対になった情報のことを指し、静的なものから動的なものまでさまざまな種類が存在する。そのような空間情報は、複数のデータを組み合わせることで単一のデータでは認識できない空間情報を捉え、アプリケーションに活用することができる。例えば、ナビゲーションシステムは、国土地理院の「基盤地図」や「衛星写真」のような地図の基盤となる空間情報の上に、GNSS (Global Navigation Satellite System/ 全球測位衛星システム) のような位置検出システムによって得られる移動体の位置や、目的地を表す POI (Point of Interest) といった空間情報を重ねることで目的地までの経路探索を行う複数の空間情報を活用したアプリケーションである。

このような複数の空間情報を活用するアプリケーションを実現する場合、アプリケーション開発者は、必要な空間情報を全て収集し、データ間のシンタックスやセマンティックスの違いに対応するためのデータ処理・分析をアプリケーション側で実現する必要がある。

本研究では、空間情報活用のためのデータ統合基盤を提案する。この基盤は、大きく分けて「データ収集」「データ処理・分析」「データ提供」の3つの機能を有している。「データ収集」機能では、さまざまなデータ提供者の所有するデータを収集し、基盤上の空間データベースへの登録を行う。基盤に収集されたデータは、アプリケーション開発を行うユーザの要求に応じて「データ提供」機能を介して提供される。この際、ユーザは必要に応じて「データ処理・分析」機能を設定することが可能であり、複数のデータ処理・分析手順の出力データを取得することが可能となる。

ユーザが取得したデータの利用方法は、二通りが考えられる。一つ目は、取得したデータを自身のアプリケーションに活用すること、二つ目は、取得したデータを必要に応じて加工し、再度データ統合基盤に登録することである。この二つ目の利用方法によって、同様のデータの活用を検討している他のユーザが、そのデータを作成するためのデータ処理・分析手順を介さずにデータを取得することが可能となる。

提案するデータ統合基盤の機能を用いてデータを取得することで、アプリケーション側で行う必要があった個々のデータの収集とデータ処理・分析手順を削減することができるだけでなく、アプリケーション側のデータやデータ処理・分析機能に要するリソースを削減することを可能とする。また、このデータ統合基盤は単一のデータソースとデータ処理・分析機能を複数のユーザに共通で提供するため、アプリケーション間でデータやデータ処理・分析機能を重複して保持することを減らすことができる。

本研究では、ユーザが基盤からデータを取得するための API を実装し、「バス待ち人数の把握」、「ドローン飛行可能空域の把握」という二つのシナリオに対して動作確認を行った。その結果、一つ目のシナリオにおいて、アプリケーション側のデータ収集とデータ処理・分析手順を 60%，アプリケーション側が保持するデータ量を 99.8%，同様に、二つ目のシナリオにおいて、データ収集とデータ処理・分析手順を 90%，データ量を 75%削減することが確認できた。

目次

第1章	はじめに	1
1.1	研究背景	1
1.2	研究目的	2
1.3	本論文の構成	2
第2章	空間情報の現状	4
2.1	空間情報の種類	4
2.1.1	静的な空間情報	4
2.1.2	動的な空間情報	6
2.1.3	データの分類	7
2.2	データ処理・分析	8
2.2.1	共通データ処理・分析	8
2.2.2	ベクタデータ処理・分析	10
2.2.3	ラスタデータ処理・分析	13
2.2.4	3D データ処理	14
2.2.5	その他のデータ処理・分析	14
2.3	活用分野	15
2.3.1	まちづくり	15
2.3.2	防災	15
2.3.3	交通・物流	16
2.3.4	農業	16
2.3.5	その他	17
2.4	関連技術	17
2.4.1	GIS (Geographic Information System)	17
2.4.2	SDI (Spatial Data Infrastructure)	19
第3章	システム提案	21
3.1	システム概要	21
3.2	要求分析	22

3.2.1 ユーザ	22
3.2.2 機能要求	22
3.3 システム設計	24
3.3.1 機能	24
3.3.2 データモデル	25
3.3.3 「データ取得」API	26
第4章 実装	30
4.1 システム構築	30
4.2 「データ取得」API	32
第5章 実験	38
5.1 実験目的と方法	38
5.2 実験環境	38
5.3 シナリオを用いた動作確認・比較評価	39
5.3.1 シナリオ1：バス待ち人数の把握	39
5.3.2 シナリオ2：ドローン飛行可能空域の把握	42
5.4 応答時間	45
5.4.1 シナリオの応答時間	46
5.4.2 データサイズと応答時間	47
第6章 議論	49
6.1 APIの設計	49
6.2 データ処理・分析手順の実現方法	50
6.3 データベース	51
第7章 おわりに	53
謝辞	55
本研究に関する对外発表	56
参考文献	57

図 目 次

1.1	空間情報活用の現状（概念図）	2
2.1	ポイントデータ	7
2.2	ラインデータ	7
2.3	ポリゴンデータ	7
2.4	SDI の概念図	20
3.1	データ統合基盤の概念図	21
3.2	データ統合基盤の利用方法（ユースケース図）	24
3.3	データ統合基盤の機能一覧	25
3.4	空間データモデル	26
3.5	空間データのテーブルイメージ	26
3.6	データ処理・分析手順の分類	27
3.7	circle 型の空間検索	29
3.8	bbox 型の空間検索	29
4.1	システムの配置図	30
4.2	概念データモデルと物理データモデルの対応	31
4.3	「データ取得」のシーケンス図	36
5.1	データ処理・分析手順のイメージ（出典：国土地理院地図を加工して作成）	40
5.2	シナリオ 1 のデータ処理・分析手順	40
5.3	データ処理・分析手順のイメージ（出典：国土地理院地図を加工して作成）	43
5.4	シナリオ 2 のデータ処理・分析手順	44
5.5	一部を事前に実行済みのシナリオ 2 のデータ処理・分析手順	47
5.6	データ取得（処理・分析なし）の平均応答時間	48
6.1	データ処理・分析手順の実現方法のイメージ	50

表 目 次

2.1	空間データの分類	8
2.2	座標系と EPSG コード	9
2.3	GIS のカテゴリ	18
3.1	クエリパラメータ	28
3.2	空間範囲条件の表現方法	28
3.3	時間範囲条件の表現方法	29
4.1	データ処理・分析名の一覧	35
5.1	実験環境	38
5.2	Python ライブラリとデータベース	38
5.3	前提のデータ	39
5.4	結果	42
5.5	前提のデータ	43
5.6	シナリオ 2 の結果	45
5.7	シナリオの応答時間	46
5.8	各シナリオの各データ処理・分析の計算時間	46
5.9	テストデータ	47

第1章 はじめに

本章では、本研究の背景、目的、本文の構成について述べる。

1.1 研究背景

空間情報とは、実世界におけるモノの位置情報とそれに関連づけられた位置以外の属性情報のことを指し、静的なものから動的なものまでさまざまな種類が存在する。例えば、電子地図における位置の基準となる基盤地図情報や、地形、建物、植生、インフラ、景観などを3次元で表現する3次元都市モデルは、地図を作成する際の基盤的な空間情報としてさまざまなアプリケーションに活用されている。また、これらの基盤的な空間情報の他にも、市区町村が住民全体の住民票を世帯ごとに編成し作成する公簿を町丁目別に集計した情報である住民基本台帳データや、各自治体の避難施設、バス停などの特定の意味を持つPOI(Point Of Interest)のようなデータも静的な空間情報の一種と考えることができる。さらに、自然現象や人工物に関する情報を表現するセンサーデータや、それらのデータの解析によって得られる人流データなども動的な空間情報として今後のビジネスを牽引していく可能性がある [1]。空間データとは、これらの空間情報をコンピュータ処理を前提として表現したデータのことを指す。

このような空間情報をさまざまな分野のアプリケーションに活用するために、データの作成、管理、処理、分析などに関して、地理情報科学に関する多くの研究や開発がなされてきた。地理情報システム (GIS: Geographic Information System) は、地理空間情報の作成、管理、表示、検索などの機能を有するシステムであり、デスクトップ型のものや、一部機能がコンポーネント化されたライブラリとしてのGISが様々な分野で利用されている。ユーザは、それらのGISのデータ処理・分析機能を用いてさまざまな種類の空間データ間のセマンティックスやシンタックスの違いに対応し、アプリケーションに必要な空間情報を導出することができる。

また、データの収集に関しては、空間情報を活用するアプリケーションの開発者は、各データ保有者から直接、または、ポータルサイトから必要なデータを収集する必要がある。

図 1.1 は、このような空間情報活用の現状を表す概念図である。

このように、空間データの作成、管理、処理・分析、共有などに関して、GIS や SDI を中心にさまざまな技術が開発されてきたが、空間情報を活用するアプリケーション開発には、依然として以下のような点が課題として挙げられる。

- 全てのデータの収集、データ処理・分析手順に開発時間が費やされている
- アプリケーションに活用する空間情報を導出するための元となるデータを全て収集・保持する必要がある、アプリケーション側のリソース負荷につながる
- アプリケーションに活用する空間情報を導出するためのデータ処理・分析機能を保持する必要がある、アプリケーション側のリソース負荷につながる

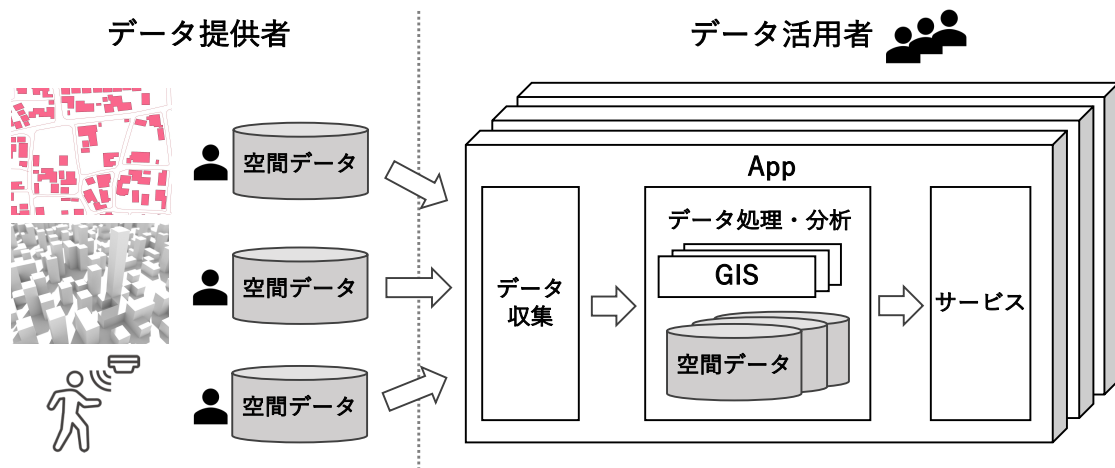


図 1.1: 空間情報活用の現状（概念図）

1.2 研究目的

本研究では、空間情報活用のためのデータ統合基盤として、さまざまな種類の空間データを収集し、要求されたデータ処理・分析手順を実行した結果のデータを提供する API を提案する。これにより、アプリケーション側のデータの収集とデータ処理・分析手順を削減し、アプリケーション側のデータやデータ処理・分析機能に要するリソースを削減することを目的とする。

1.3 本論文の構成

本論文は、本章を含めて7章で構成される。構成は以下のようになっている。

- 第1章
 - － 研究背景と目的，本論文の構成について述べる．
- 第2章
 - － 本研究で提案するシステムの機能を検討するにあたり調査した空間情報の現状について述べる．
- 第3章
 - － 本研究で提案するシステムについて述べる．
- 第4章
 - － 本研究で提案するシステムの実装について述べる．
- 第5章
 - － 提案システムの動作確認とその結果について述べる．
- 第6章
 - － 提案システムの設計に関して検討すべき点について述べる．
- 第7章
 - － 本論文のまとめ，今後の課題について述べる．

第2章 空間情報の現状

本章では、空間情報の現状として、以下の4項目について述べる。

1. どのような種類のデータが存在するか
2. 複数種類のデータの横断的な活用のためにどのような処理・分析を要するか
3. どのような活用分野が存在するか
4. 関連技術（GIS, SDI）

2.1 空間情報の種類

ここでは、現在活用されている代表的な空間データの種類について、静的なものとの動的なものについて分けて述べる。

2.1.1 静的な空間情報

基盤地図

国土地理院が中心となって整備を進めている電子地図における位置の基準となる情報である。整備項目には「測量の基準点」「海岸線」「公共施設の境界線（道路区域界）」「公共施設の境界線（河川区域界）」「行政区画の境界線及び代表点」「道路縁」「河川堤防の表法肩の法線」「軌道の中心線」「標高点」「水涯線」「建築物の外周線」「市町村の町若しくは字の境界線及び代表点」「街区の境界線及び代表点」の13項目が含まれる。空間情報を可視化する際の土台となる地図として利用される。

3D都市モデル（CityGML）

地形、建物、植生、道路などのモデルを含む3次元の空間情報によって都市環境を表現したものである [3]。日本では、Project PLATEAU として国土交通省が

3D 都市モデルの整備・活用・オープンデータ化を進めている。データの詳細度は LOD(Level of Detail) として 5 段階で表現されており、LOD0 では建物の高さ、LOD4 では建物の内部構造まで含まれる [4]。

国土数値

地形、土地利用、公共施設、道路などの国土に関する基礎的な空間情報であり、オープンデータとして無償で提供されている。国土（水、土地）、政策区域、地域、交通、位置参照情報の分類でさまざまな空間情報が存在する。

POI (Point Of Interest)

Point of Interest (POI) は、意味を持つ特定の場所や目標物のことを指す（避難所、店舗、トイレ、AED 設置位置など）。

BIM/CIM

Building Information Modeling(BIM) は、建築物のライフサイクルを通じて、設計やプロジェクトに関わる重要なデータをデジタル形式で管理する方法論のことであり [5]、計画、設計、施工、維持管理などにおいて 3 次元建物情報モデルデータが BIM データとして活用されている。点群とは異なり、建物の形状だけでなく、材質、メーカー情報などの属性を扱うことができる。BIM が建築分野のデータを扱うのに対し、Construction Information Modeling (CIM) では土木分野で地形、土工形状、地質などのモデルも対象となる。

点群

レーザスキャナーなどの測量機器によって得られる 3 次元点群データは、位置合わせなどの加工処理を施すことで、道路、建物、植生などの地物の 3 次元モデルとして扱うことができる [6]。

住民基本台帳（統計）

もともと、氏名、生年月日、性別、住所などが記載された住民票を編成したものであるため、個々の情報を使用することはできないが、空間範囲に基づく統計データとして扱うことで、個人のプライバシーに配慮しながら、人口の統計情報

などを扱うことが可能となる。このような統計データは、地域単位での人口推移の予測などのサービスに活用することが期待できる。

筆ポリゴン（農地区画）

農林水産省が実施する耕地面積調査等の母集団情報として、全国の土地を隙間なく 200 メートル四方（北海道は 400 メートル四方）の区画に区分し、そのうち耕地が存在する約 290 万区画について衛星画像等をもとに筆ごとの形状に沿って作成した農地の区画情報のことを指す。区画情報の一元的な管理を実現するだけでなく、将来的には、自動運転、衛星測位システム及びドローン等への活用、災害状況把握などさまざまな目的に活用することが期待されている。

2.1.2 動的な空間情報

動的な空間情報には、センサやカメラ、人の位置に関する情報が含まれており、抽象度の低いものから高いものが混在する。

センサ

ある対象に関する空間情報・時間情報を収集するセンサの市場は増加傾向にあり、リアルタイムのサービスを提供する多様なアプリケーションに活用されることが期待できる。例えば、リアルタイムの交通量や気候、自然に関わる状況を把握するために、自動車を交通観測モニタリング装置と捉えたプローブカーの開発が進められている [1]。航空写真などの画像データも、地図を表示するアプリケーションなどで活用されている。

カメラ

地理情報科学の分野では、リモートセンシング技術の発達とともに、衛星写真が地図データとして幅広く活用されてきた。また、Web カメラなどによる、さまざまなものごとの監視システムが多く開発されており、街中に設置されたカメラによって河川水位、降雪量、人流、駐車場の空き状況などの空間情報を導出するために活用することが期待できる。

人

空間情報を活用するアプリケーションの多くは人に対してサービスを行う。そのため、ユーザの位置情報や、災害時の人流、混雑状況などの人に関する情報は、さまざまな分野で重要な空間情報として活用されている。全球測位衛星システム (GNSS) や、スマートフォンの普及により、マップサービスのユーザは、現在位置をリアルタイムに把握することが可能となった。これにより、目的地までのナビゲーションシステムなど、ユーザの位置に基づくさまざまなサービスが開発されている。

2.1.3 データの分類

前述した空間情報の種類では、コンピュータでの処理を想定した情報から、「人」のような抽象的な情報まで幅広く紹介した。抽象的な空間情報に関してはコンピュータで扱うために空間データ化する必要があるが、この際、多くの空間情報に対して適応できるデータモデルが GIS データモデルである。GIS データモデルには「ベクタデータ」「ラスタデータ」「3D データ」の分類が存在する。

ベクタデータとは、地物（フィーチャ）をポイント、ライン、ポリゴンで表現したデータのことを指す（図 2.1, 2.2, 2.3）。前述した空間情報の中でも、基盤地図、国土数値、POI、筆ポリゴン、センサ、人など、多くの種類がこのベクタデータとして表現されている。現実世界に存在する地物や市町村境界など、明確な境界を持つを表すものに利用される。一方で、境界が曖昧な気象などといったものには適していない。そのようなデータにはラスタデータが利用される。

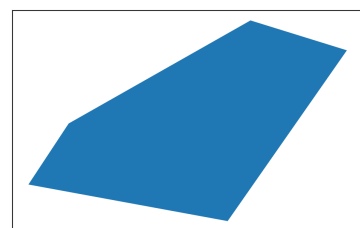
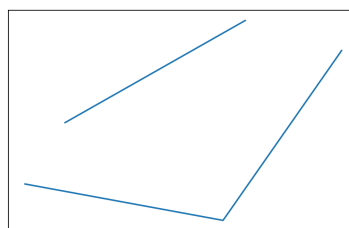
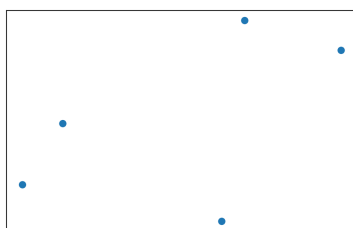


図 2.1: ポイントデータ

図 2.2: ラインデータ

図 2.3: ポリゴンデータ

ラスタデータとは、格子状に並んだセルで構成されるデータのことを指す。前述したデータの中でも、基盤地図の標高モデルやカメラによる衛星写真がこのデータに該当する。素早く描画できるといった特徴があり、背景などに利用される一方で、データの粒度がセルの大きさによって決まるという特徴があるため、明確な境界を持つ市町村境界などを表すデータには適していない。そのようなデータにはベクタデータが利用される。

3D データとは、高さを加えた3次元の座標を持つデータのことを指す。前述したデータの中でも、3D 都市モデルや BIM/CIM、点群がこのデータに該当する。高さが加わることで、より詳細に実世界を表現することができる。3D 都市モデルのような基本形から BIM/CIM、点群系のようにさまざまなデータモデルが存在するため、それぞれのデータモデルに適した管理が必要となる。

表 2.1 は、前述した空間情報が該当する空間データの分類を示す。

表 2.1: 空間データの分類

データモデル	データ	静的 or 動的
ベクタ	・ 基盤地図 ・ 国土数値 ・ 農地区画（筆ポリゴン） ・ POI（バス停、避難施設など） ・ 人（人流、混雑状況など） ・ センサ（温湿度、測位など）	静的 静的 静的 静的&動的 動的 動的
ラスタ	・ 標高モデル ・ 衛星写真	静的 静的&動的
3D ベクタ	・ 3D 都市モデル（CityGML）	静的
その他	・ 住民基本台帳（統計） ・ BIM/CIM ・ 点群 ・ カメラ（水位、降雪量、人流など）	静的 静的 静的&動的 動的

2.2 データ処理・分析

ここまで、空間情報の種類について述べてきたが、実際にデータをさまざまな分野のアプリケーションに活用する場合は、データ間のシンタックスやセマンティックスの違いに対応するためのさまざまなデータ処理・分析が必要となる。

ここでは、既存の空間データを活用したり、複数の空間データから新しく高次データを導出するために必要となるデータ処理・分析について述べる。

2.2.1 共通データ処理・分析

ジオプロセッシングとは、GIS データと呼ばれるベクタデータ・ラスタデータに対して何かしらの処理を行い、新しいデータを出力する一連の流れのことを指

す。一般的には GIS には、さまざまなジオプロセッシングツールが組み込まれており、それらを用いてデータを加工することが可能となる。

ここでは、一般的な GIS で利用されるジオプロセッシングについて述べる。

座標参照系変換

GIS では、地球上の特定の位置を示すために座標が使用される。座標は原点を基準とした位置で表現したものである。QGIS では座標参照系 (CRS : Coordinate Reference System) という言葉で表現される。空間情報の座標参照系には、「地理座標系」と「投影座標系」の二つが存在する。

地理座標系とは、地球の重心を原点とし、座標の単位を緯度と経度という角度で表現する座標系のことを指す。地球の形状を表現する楕円体の定義がさまざまなため、国によって採用している楕円体の形状や重心位置の基準が異なる。そのため、地理座標系にはいくつかの種類が存在する。

投影座標系とは、3次元である地球を2次元の平面に投影し、XY 座標で表現する座標系のことを指す。この際の投影方法を、地図投影法と呼び、用いる投影法や設定する原点などの違いによって、さまざまな投影座標系が存在する。日本では、全世界を経度6度ごとのゾーンに分けて東回りに番号を付けて規格化した UTM 座標系や、公共測量で採用されている平面直角座標系がある。投影座標系では地球の形状なども考慮して投影を行うため、地理座標系の定義も含まれる。

このように、空間情報の座標系にはさまざまな種類が存在するため、複数の空間情報を重ね合わせた処理や可視化を行う場合、座標系変換が必要となる。この際、座標系を識別するために使用されるユニークな ID が「EPSG (European Petroleum Survey Group) コード」である。表 2.2 では、日本で一般的に使用される座標系と EPSG コードの対応を示す。

年	座標系	EPSG コード
2011	地理座標系 (JGD2011)	6668
	UTM 座標系	6688～6692
	平面直角座標	6669～6687
2000	地理座標系 (JGD2000)	4612
	UTM 座標系	3097～3101
	平面直角座標	2443～2461
	地理座標系 (WGS84 系)	4326

表 2.2: 座標系と EPSG コード

ジオコーディング

空間情報に対するジオコーディングとは、住所、地名、目標物、郵便番号などが指す場所に対して、座標を与えることを指す。この際、場所と座標の対応表として位置参照情報が必要となる。日本では、街区レベルの位置参照情報を国土交通省が整備している。

2.2.2 ベクタデータ処理・分析

空間構成単位の変換

空間情報には、分析に適した解像度・空間構成単位に変換するための集約（aggregation）や単純化（simplification）といった処理がしばしば行われる。これらの処理は、データを軽量化する効果がある。

集約とは、データの粒度が必要以上に大きい場合に、ある領域ないでの均一性を仮定してデータを集計する変換のことを指す。GISでは、融合（dissolve）という機能を用いて集約が実現できる。この処理は、ラスタデータに行われる処理として定義されることが多いが、ベクタデータに対しても融合（dissolve）という機能を用いて集約と同様の機能を実現することが可能である。データ量を抑えられるだけでなく、プライバシーの問題を保護する際にも有効である [7]。

単純化とは、多くの頂点で構成された複雑なポリゴンやラインデータに対してより少数の頂点で構成される単純な図形に変換することを指す。代表的なアルゴリズムとして Douglas-Peucker のアルゴリズムが存在する。手順は以下の通りである。

1. 許容範囲となる閾値を決定する
2. 両端点を結んだ直線とその他の頂点との距離を垂直方向に計測する
3. 閾値より遠い頂点が存在する場合に、最も遠い頂点によってラインを分割する
4. 分割した各ラインに 1,2 の処理を同様に行い、閾値の範囲内にある全ての頂点が除外されるまで処理を継続する

空間解析

地理情報科学の分野では、基本的な空間解析として、基本量の測定、オブジェクト選択（抽出）、オーバーレイ、その他データ操作といった処理が頻繁に行われ

ている。

基本量の測定は、ベクタデータの中でもラインデータの全長や、ポリゴンデータの面積などを導出する操作である。それらの値を導くためには、空間情報がXY座標で表現された投影座標系である必要があるため、必要に応じて座標系変換が事前に行われる。

オブジェクト選択（抽出）は、空間データを構成する複数オブジェクトの中から、特定の空間データを抽出する操作である。最も簡単な方法はGISのGUI (Graphical User Interface) を利用して、図形またはテーブルのレコードを選択することであるが、空間的特徴に基づく空間検索や、属性値に基づき条件に該当するオブジェクトを選択する属性検索なども有効な方法である。

オーバーレイは、二つ以上の空間データを重ね合わせることで、特定の条件を満たす領域を抽出する操作である。具体的な条件としては、以下のようなものが存在する。

- union : 複数の空間データの結合する
- intersect : 複数の空間データが重なる部分のみを抽出する
- symetric difference : いずれかの空間データにのみ含まれる（重ならない）部分のみを抽出する
- erase(difference) : ある空間データから別の空間データと重なる部分を削除する
- identify : ある空間データを別の空間データとの重なりによって分割する

最後に、その他データ操作には merge, clip と呼ばれる操作が存在する。それぞれの操作は以下になる。

- merge : 同じフィールド（列）を持つ複数の空間データを結合する
- clip : ある空間データから切り抜く空間領域のみを切り出す

領域分析

地理情報科学における領域分析とは、ある地物について、それと関わりのある空間範囲を捉えようとするものである [7]。代表的なものとして、バッファによる領域分析とボロノイ領域による領域分析が存在する。

バッファとは、一般的にあるオブジェクトからの距離が特定の距離以下である領域のことを指す。例えば、オブジェクトが点であれば、バッファは円となる。新しい鉄道駅や商業店舗の位置を決定する際などにしばしば利用される。

ボロノイ領域とは、複数のオブジェクトが存在する場合に、各オブジェクトについて、そのオブジェクトまでの距離が他のオブジェクトまでの距離よりも小さい領域のことを指す。バッファと同様、各鉄道駅や各商業店舗が最寄りとなる領域を導出することが可能であり、売り上げの予想などに活用することができる。

ネットワーク分析

地球上の地物とその関係をグラフで表現し、頂点や変に具体的な意味づけを行うことで、さまざまな情報の導出を試みることをネットワーク分析と呼ぶ。代表的なものとして、ユーザが道路ネットワークを介して目的地に向かう際に、最小コストで辿り着ける経路や最短時間で辿り着ける経路を発見する最短経路探索が存在する。この分析は、移動体を扱うさまざまな分野のアプリケーションで頻繁に利用される。

ポイントデータの分析

空間情報の中でも POI、人、センサに関するものはベクタの点データに分類されるデータが多く存在する。これらのポイントデータの分布に関する分析は、人口、都市施設、動植物、化学物質、湖沼、遺跡など分野毎の対象に対して行われている。ポイントデータの分析手法は大きくわけて、視覚的分析と数理的的分析が存在する。

視覚的分析では、可視化によって点が密集している地域を把握したり、メッシュや行政区域によって点を集計したりことで、点分布の特徴を説明する。

数理的的分析では、解釈が主観的になりがちである視覚的分析の短所を補うために、点分布の特徴を表す数値指標を定義し、その大小などによって特徴を説明する。利用される数値指標には以下のようなものが存在する。 n は点の総数を表す。

- 平均最近隣距離：全ての点に対して最寄りの点までの距離 d_i の平均値

$$W = \frac{1}{n} \sum_{i=1}^n d_i \quad (2.1)$$

- Ripley の K 関数：各点までの距離 k が距離 d 以内にある他の全ての点の個数

$$L(d) = \sqrt{\frac{A \sum_{i=1}^n \sum_{j=1, i \neq j}^n k_{ij}}{\pi n(n-1)}} \quad (2.2)$$

- 標準距離：各点の位置 (x_i, y_i) が点データの重心からどの程度広がっているかを表す指標

$$r = \frac{\sum_i x_i^2 + y_i^2}{n-2} \quad (2.3)$$

- 標準楕円距離：長径 r_1 ・短径 r_2 ・Y 軸からの時計回り方向の傾き θ によって点がどの方向に広がっているかを表す指標

$$\begin{aligned} \theta &= \tan^{-1} \left(\frac{\sum_i x_i^2 - \sum_i y_i^2 + \sqrt{(\sum_i x_i^2 - \sum_i y_i^2)^2 + 4 \sum_i x_i^2 y_i^2}}{2 \sum_i x_i y_i} \right) \\ r_1 &= \sqrt{\frac{2 \sum_i (x_i \cos \theta - y_i \sin \theta)^2}{n-2}} \\ r_2 &= \sqrt{\frac{2 \sum_i (x_i \sin \theta + y_i \cos \theta)^2}{n-2}} \end{aligned} \quad (2.4)$$

2.2.3 ラスタデータ処理・分析

フィルタリング

近傍のセルに与えられた数値を考慮してセルの値を変更し、異なるデータを作成するデータ処理。対象となるセルと処理に用いる近傍のセルを含む範囲は移動窓（フィルタ）と呼ばれ、通常は 3×3 の 9 セルで構成される。

ラスタ演算

複数のラスタデータの値を用いた計算を行い、新しいラスタデータを作成するデータ処理。

流域分析

「水は高きより低きに流れる」という原理のもと、あるセルから水が流水する方向は、周囲 8 セルのうち最も低いものであるという単純な仮定を用いて全てのセルの流下方向を定義するデータ分析。基礎となるデータとして、DEM (Digital

Elevation Model：数値標高モデル）を用いて地表を水がどのように流れるかを再現する。

2.2.4 3D データ処理

点群処理

現在、LiDAR などの技術の発展に伴い、点群がさまざまな場面で利用されている。点群を活用するために、位置合わせ、ノイズ処理といった前処理に加えて、干渉チェックなどの解析を行った上で、3D モデルなどに変換される。

2.2.5 その他のデータ処理・分析

ベクタ・ラスタデータなどの位置情報が付与されているデータは GIS で扱うことができるデータとして頻繁に活用されている。一方で、センシング技術などの発展によってカメラや各種センサから得られるリアルタイムのデータは位置情報を含んでいない形式でデータが共有されている場合があるものの、さまざまな分野で活用可能な空間情報を潜在的に含んでいる可能性がある。

ここでは、カメラやセンサのデータから空間情報を導出するために必要なデータ処理・分析について述べる。

GIS データ化

センサデータは、一般的に csv 形式や json 形式で管理される。これらのデータはセンサの設置位置という空間表現を付与することで GIS データにすることが可能となる。

画像データに関しても同様である。現在、地域の状況を把握するためなどの監視カメラの普及が進んでいる。それらのカメラから得られる画像データには、潜在的に人流や交通流に関する空間情報を含んでいるため、適切なデータ処理・分析によってそれらの空間情報を導出し、GIS データ化することで、アプリケーションに活用することが考えられる。

時系列データ処理

GIS データ化されたセンサデータや、人流・交通流データは短い時間で属性値や座標値が変化するデータである。このようなデータには、時系列データ処理に

よって、高次データを生成できることが考えられる。例えば、一定時間のデータから将来のセンサ値や座標を予測することが考えられる。このような処理を一般的に利用することを検討するためには、センサデータのような属性値が頻繁に変化するデータや、人流・交通流データのような座標値が頻繁に変化するデータそれぞれに対してデータモデルの標準化を行い、データ処理・分析をコンポーネント化する必要がある。例えば、Asahara らは移動体データの標準化に関する研究を行なっている [8]。

2.3 活用分野

空間情報には前述したようにさまざまな種類が存在するが、これらのデータは活用する目的に応じて作成される。ここでは、作成された空間情報が各分野でどのように活用されているかを述べる。

2.3.1 まちづくり

まちづくりの分野では、都市計画の策定・運用、公共空間やインフラ整備のために空間情報が活用される。基盤的な空間情報として、基盤地図や 3D 都市モデルが利用されており、特に国土交通省が主導している Project PLATEAU に基づき、LOD1 または LOD2 の CityGML 形式の 3D 都市モデルの作成が各地で進められている。基盤地図の上には各種国土数値や筆ポリゴン、公共施設の POI などの静的な空間情報を重ねることで、都市の土地利用状況を把握することが期待できる。また、各種センサを都市に配置し、リアルタイムの人流や交通流といった動的な空間情報を作成することで、公共空間の利用状況・混雑状況の把握や、土地利用計画・道路計画の策定への空間情報の活用が可能となる。

上述したデータを統合し、各種アプリケーションのサービスを実現するためには、ベクタデータに対する「座標系変換」「空間解析」「領域分析」や、センサデータや統計データに対する「時系列データ処理」といったようなデータ処理・分析が必要となる。

2.3.2 防災

防災分野では、ハザードマップに始まり、災害対策計画、気象警報・避難指示、被害推定、被害状況の把握、被災者支援、復興計画といった災害マネジメントサイクル全体で空間情報が必要となる。

ハザードマップの作成には基盤地図の標高モデル、災害対策計画にはそれに加えて3D都市モデルや人（人口分布や人流データ）、そして気象警報・避難指示や被害推定、被害状況の把握には各種センサデータが活用される。さらに、災害後の被災者支援や復興計画には、台帳データをもとに被災者台帳が作成・活用される。

上述したデータを統合し、各種アプリケーションのサービスを実現するためには、ベクタデータに対する「座標系変換」「空間解析」「ネットワーク分析」「領域分析」や、センサデータや人のデータ（人流）に対する「時系列データ処理」、統計データに対する「ジオコーディング」や「位置情報の付与」などのデータ処理・分析が必要となる。

2.3.3 交通・物流

交通・物流の分野では、公共交通の混雑予測や配送経路の最適化などに空間情報が活用される。活用されるデータには、静的なデータとして道路やPOI、動的なデータとして交通流、人や配送車の位置情報などが含まれる。

上述したデータを統合し、交通・物流分野のアプリケーションのサービスを実現するためには、主にベクタデータに対する「座標系変換」「空間解析」「ネットワーク分析」などのデータ処理・分析が必要となる。

自動車産業の分野では、LiDARやカメラなどの車載センサによるリアルタイムの点群データの取得・活用が期待されている。この点群データは、人の位置情報などの潜在的な情報を保持している可能性があるため、他のデータと統合したり、適切なデータ処理・分析を行うことで、他のアプリケーションに活用することが期待されている。

また、現在は、3D都市モデルや屋内測位技術の発達により、屋外だけでなく屋内の移動体のリアルタイムの把握が期待されている。例えば、屋内における移動体の位置情報と屋内施設の詳細地図データを組み合わせることで、屋内で動作する配送用ロボットの自動制御などが期待できる。

2.3.4 農業

農業の分野では、トラクターなどを含む農業機械の自動制御や耕地面積の推移などを把握するために空間情報が活用される。活用されるデータには、基盤的な空間情報として衛星写真や基盤地図、動的な空間情報として農業機械の位置情報などが挙げられる。また、農林水産省の整備した筆ポリゴンは耕地面積調査等の対地標本実測調査の母集団情報として農地の区画を把握することができるデータである。

上述したデータを統合し、この分野のアプリケーションを実現するためには、主にベクタデータに対する「座標系変換」「空間解析」「ネットワーク分析」などのデータ処理・分析が必要となる。

2.3.5 その他

その他にも、観光、健康・福祉、環境・エネルギー、建設業などさまざまな分野のアプリケーションに対して空間情報が必要となる。特に、3D 都市モデルは CityGML の開発により、これまでにないさまざまなユースケースが実現できると考えられている。例えば、健康・福祉の分野では、バリアフリーに対応した経路によるユーザのナビゲーションを行うアプリケーションが検討されている。他にも、環境・エネルギー分野では、3D 都市モデルによって建物や農地の日照シミュレーションを行うことが検討されている。

また、建設業の分野では、材質などを含む 3 次元の空間情報として BIM の開発と普及が進められており、今後さらなる活用領域の拡大が期待されている。

2.4 関連技術

空間情報を活用するアプリケーションを実現するためには、前述したさまざまな種類のデータから必要なデータを発見/作成/収集する必要がある。また、データに対して適切なデータ処理・分析を行うことで、アプリケーションを実現する必要がある。

地理情報科学の分野では、それらの機能を実現するために、さまざまな技術が開発されてきた。ここでは、GIS、SDI の技術について紹介する。

2.4.1 GIS (Geographic Information System)

GIS の分類

地理情報システム (GIS : Geographic Information System) とは、空間データの作成、管理、処理・分析・地図化 (表示) の機能を有するシステムであり、現在もさまざまなユーザや組織に利用されている。Steiniger ら [9] によると、GIS ソフトウェアは「デスクトップ GIS」「空間データベースシステム (Spatial DBMS)」「WebMap サーバ」「サーバ GIS」「WebGIS クライアント」「モバイル GIS」「GIS ライブラリ」に分類され、それぞれ異なる目的のために用いられる。表 2.3 はそれぞれの GIS の主な機能とソフトウェアを示したものである。

カテゴリ	主な機能	主なソフトウェア
デスクトップ GIS[10]	- データの表示 - データの作成, 編集 - データ処理・分析 - 地図の生成	- QGIS - ArcGIS
空間 DBMS	- データの保存 - データ処理・分析	- PostGIS - SpatialLite
WebMap サーバ	- Web 上でのデータの表示 - Web 上でのデータの作成, 編集 - Web 上での地図の生成	- MapServer[11] - GeoServer
サーバ GIS	- 空間データの保存 - データ処理・分析	ArcGIS Server
WebGIS クライアント	- Web 上でのデータの表示 - Web 上でのデータの作成, 編集 - Web 上でのデータ処理・分析 - Web 上での地図の生成	- Google Maps - Google Earth
GIS ライブラリ	- データの作成, 編集 - データ処理・分析 - 地図の生成	- GDAL/OGR - PROJ - GEOS

表 2.3: GIS のカテゴリ

WebGIS

これまで、空間情報を活用する産業および研究では、デスクトップ GIS で局所的に空間分析を行うことが主な方法であったが、ウェブ上で空間情報にアクセスし、処理を行う技術の開発とそれを用いた活用方法がいくつか検討されている [12]. 例えば、Open Geospatial Consortium (OGC) によって標準化された空間データに対するさまざまな機能は、いくつかの Web アプリケーションに用いられている [13, 14]. OGC の標準規格には空間データベースから地図画像を要求する HTTP インターフェースを提供する OpenGIS Web Map Service Interface Standard (WMS) や、データの発見機能を有する Catalog Service for the Web (CSW)、データの検索機能を有する Web Feature Service (WFS)、Web Coverage Service (WCS) などが存在する.

これらの、OGC の標準規格は、OGC Web Services (OWS) と呼ばれており、地理空間産業および研究の状況は、デスクトップ GIS で局所的に空間分析を行うこと

から、ウェブ上で空間データにアクセスして処理することへと変化している [12]. Web を介して空間データに対する機能を提供することによって、データ活用者は大量のストレージや高い処理能力を持つ環境を用意することなくデータに対する処理を行うことが期待できる [2].

2.4.2 SDI (Spatial Data Infrastructure)

前述した WebGIS の技術を用いてデータのアクセシビリティや再利用を促進させるフレームワークとして、空間データ基盤 (SDI: Spatial Data Infrastructure) が提案されている. SDI とは、空間データを活用するための技術、ポリシー、および異なるビジネス・インターフェイス間での空間データおよび関連情報の交換をサポートするフレームワークのことを指し、国レベルでの開発が進められている. SDI は、空間データの表示・照会・分析を行う「クライアントソフトウェア」、メタデータやサービス、空間データなどのリソースの発見・閲覧・照会を行う「カタログサービス」、インターネットを介したデータの配信を行う「空間データサービス」、座標参照系変換などを行う「処理サービス」、データを保存する「空間データ (リポジトリ)」、データの作成・編集を行う「GIS ソフトウェア」で構成される [15] (図 2.4). これらのソフトウェアコンポーネント間の相互作用を可能にするために、前述した OWS や ISO19115 (空間メタデータ)、W3C のデータ形式・データ伝送といった標準が必要となる.

日本では、空間データの活用を促進させる仕組みとして、G 空間情報センターが運用されている. G 空間情報センターでは、G 空間情報 (地理空間情報) 産業の活性化、新たなビジネス・サービスの創出を目的として、国、地方公共団体、大学、民間等が保有するオープンデータ、有償・無償データ、独自データなどがポータルサイトを介して提供されている. データ提供者は、ユーザ登録等の各種手続きを行い、共通連携基盤 (API) を介してデータやメタデータを登録することができる. また、ユーザは、登録されたデータをファイル形式でダウンロードして、自身のアプリケーションに活用することができるが、依然としてユーザ側の開発環境に全てのデータを保存する必要があるため、リソース負荷となっている.

しかし、この G 空間情報センターはデータポータルとしての役割を果たしているものの、SDI の概念に含まれる空間データサービス (空間検索によるデータの提供等) や処理サービスを提供するには至っていない.

Wagmenn ら [2] や Zhao ら [10] によると、このような現状を踏まえて空間データをアプリケーションで活用するためには、依然として以下のような課題が存在する.

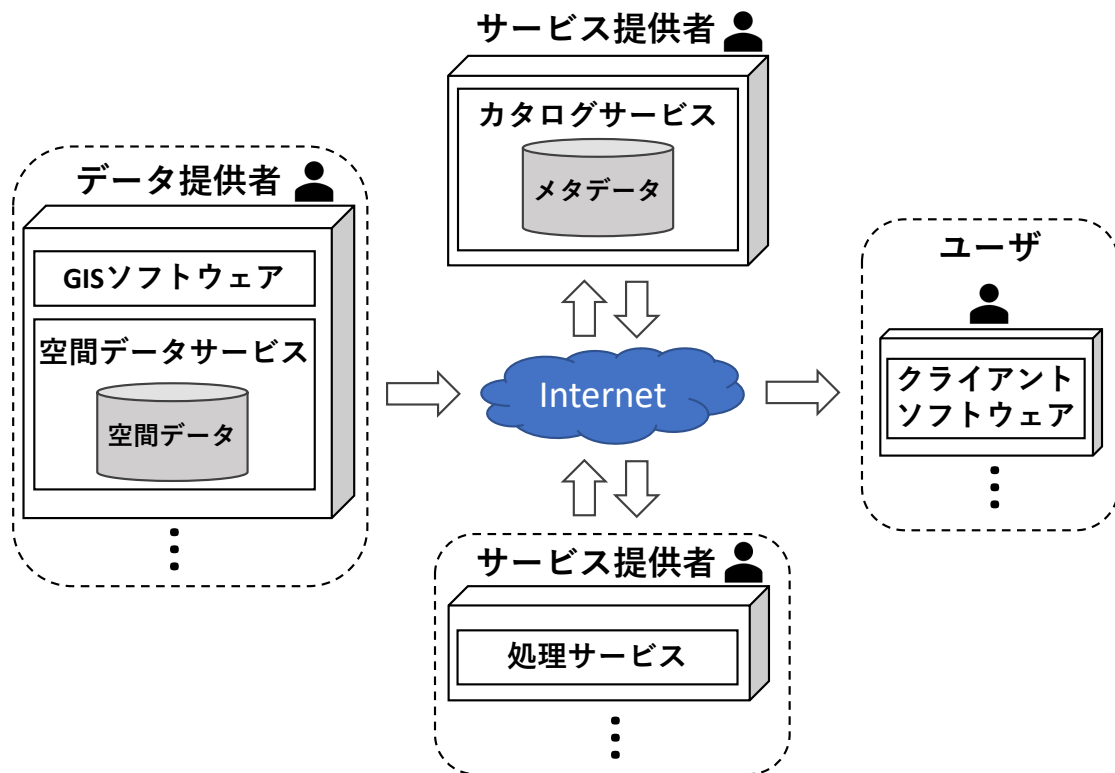


図 2.4: SDI の概念図

- データのダウンロード、データの整合化、データの前処理に多大な時間が費やされる
- 大規模な空間情報を扱うアプリケーションでは、データストレージや処理能力などのリソースを確保するために多大なコストが生じる
- データ活用者が、地理情報科学に関するバックグラウンドを持っておらず、複雑な処理・分析機能を実装することが困難な場合がある
- 従来の SDI は、静的な生データのみへのアクセスを可能にするため、動的なデータに対するアクセスの柔軟性がない

第3章 システム提案

本章では，空間情報活用のためのデータ統合基盤を提案し，その詳細について，概要，要求分析，システム設計の順に述べる．

3.1 システム概要

本研究では，空間情報活用のためのデータ統合基盤を提案する（図3.1）．このデータ統合基盤の機能は大きく分けて次の三点である．各機能の詳細は後述する．

1. データ収集
2. データ処理・分析
3. データ提供

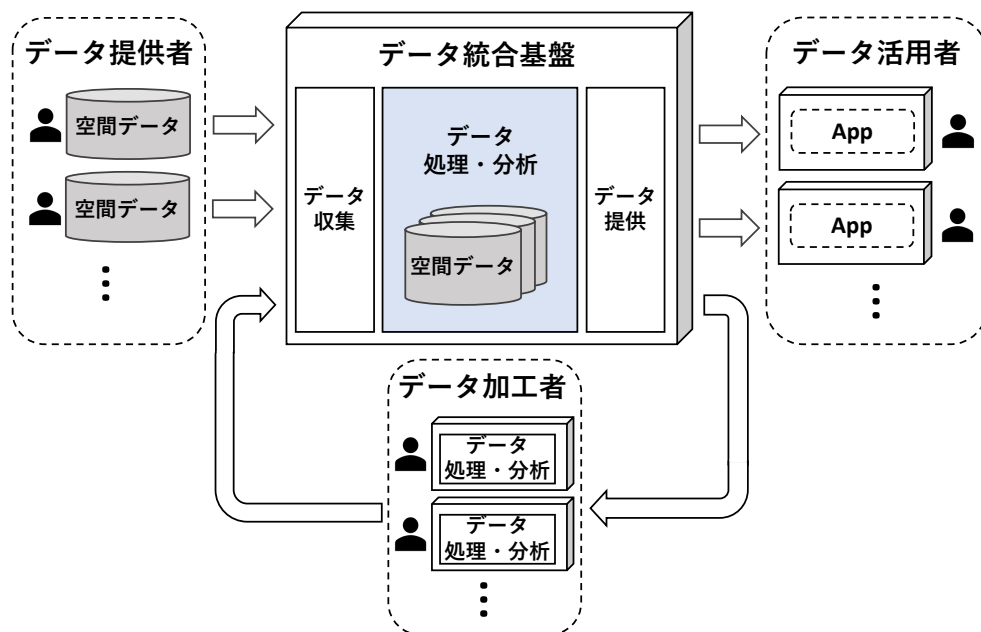


図 3.1: データ統合基盤の概念図

3.2 要求分析

提案するデータ統合基盤の詳細機能を検討するにあたり、現状の空間情報の活用方法についてまとめる。

3.2.1 ユーザ

データ統合基盤では、以下の三種類のユーザの利用を想定する（図 3.1）。

1. データ提供者

自身の保有する空間情報を基盤に提供する。

2. データ活用者

基盤に提供されたデータの中から、特定のデータを発見・取得し、アプリケーションに活用する。

3. データ加工者

基盤に提供されたデータの中から、特定のデータを発見・取得した後、必要に応じたデータ処理・分析を行うことで高次データを生成し、再度基盤に提供する。データ加工者が地理情報科学などの専門知識を要する複雑なデータ処理・分析を行なったデータを共有することで、専門知識を要さないデータ活用者が高次データを容易に活用することを可能とする。

3.2.2 機能要求

データ収集・データ提供

前述した三種類のユーザによってデータ統合基盤の利用方法が異なる。ここでは、各種類のユーザ毎にデータ統合基盤に要求される機能について検討する。

1. データ提供者

データ提供者は、自身が保有するデータをデータ統合基盤に提供し、データ活用者に発見してもらう必要がある。データの提供は、「**データ登録**」というアクティビティを実現する機能をデータ統合基盤に用意する必要がある。データの発見は、データに関するメタデータの「**カタログ登録**」というアクティビティを実現する機能をデータ統合基盤に用意する必要がある。この「**カタログ登録**」のア

クティビティを通してデータを他のユーザに発見してもらうために、「**カタログ検索**」の機能も必要となる。この機能により、各データ活用者やデータ加工者が必要なデータを発見することができるだけでなく、データ提供者がどのような情報を付与することで他のユーザが検索しやすいかを踏まえてメタデータを作成することを可能とする。

これらのアクティビティを実現する機能は、データ統合基盤側では「データ収集」機能として実現される。

2. データ活用者

データ活用者は、データ統合基盤に登録されたデータの中から、自身の開発するアプリケーションに必要なデータを発見し、取得する必要がある。データの発見は、データ提供者が登録したメタデータを用いた「**カタログ検索**」というアクティビティを実現する機能をデータ統合基盤に用意する必要がある。カタログ検索により、必要なデータを発見したデータ活用者がそのデータを取得するために、「**データ取得**」というアクティビティを実現する機能をデータ統合基盤に用意する必要がある。

これらのアクティビティを実現する機能は、データ統合基盤側では「データ提供」機能として実現される。

3. データ加工者

データ加工者は、データ統合基盤に登録されたデータを発見・取得し、処理・分析を行うことで得られる高次データを再度データ統合基盤に登録するため、データ提供者・活用者の両方のアクティビティを行う。

データ処理・分析

本研究で提案するデータ統合基盤の目的には、空間情報を活用するアプリケーション開発者が行う必要があるデータ処理・分析を、基盤上で実現することで、アプリケーション側に収集されるデータを削減すること、データ処理・分析機能に要するリソースを削減することを目的としている。

具体的に基盤上で行うデータ処理・分析としては、前章のデータ処理・分析の部分で示した中から、視覚的分析を必要としないデータ処理・分析を選択する。これらの機能を複数ユーザに同時に共通で提供することで、さまざまな分野のアプリケーション開発におけるデータとデータ処理・分析機能に要するリソースを削減することができる。

データ統合基盤の利用方法

図 3.2 は、前述したデータ収集、データ処理・分析、データ提供の機能要求を踏まえて、データ統合基盤の利用方法を示したものである。データ活用者およびデータ加工者が「データ取得」を行う場合は、データ、検索条件、データ処理・分析を指定する。ここで、データとは、カタログ検索によって発見したデータ名のことを指す。検索条件については、後述する API 設計にて詳細を説明する。また、データ処理・分析とは、指定したデータに対して行うデータ処理・分析名のことを指す。これにより、データ活用者およびデータ加工者は必要最小限のデータ処理・分析手順の出力データを取得し、アプリケーションに活用することができる。

本研究では、この中でもデータ取得に焦点を当てて設計を行う。

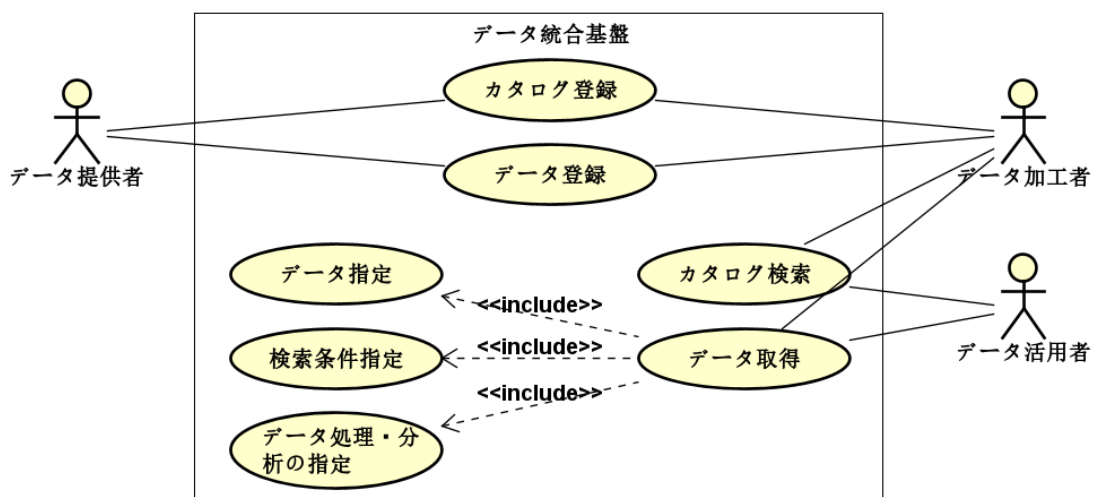


図 3.2: データ統合基盤の利用方法（ユースケース図）

3.3 システム設計

3.3.1 機能

前述した機能要求を踏まえて、データ統合基盤に要する機能をまとめたものが図 3.3 である。

データ収集、提供については、前述した機能要求で述べた機能に加えて、「契約」と「課金」についての機能を追加している。このデータ統合基盤は、多数ユーザの利用を想定してため、基盤の運用のためにこれらの機能が必要となる。「契約」については、具体的にデータ提供者識別のための「ID 付与」、データ通信のための

「プロトコル設定」、認証手続きのための「トラストアンカー」といった機能が必要となる。「課金」については、登録するデータに対する「価格設定」の機能が必要となる。この機能により、データ提供者が利益を得られる体制を整え、データ登録の利用を促すことができる。

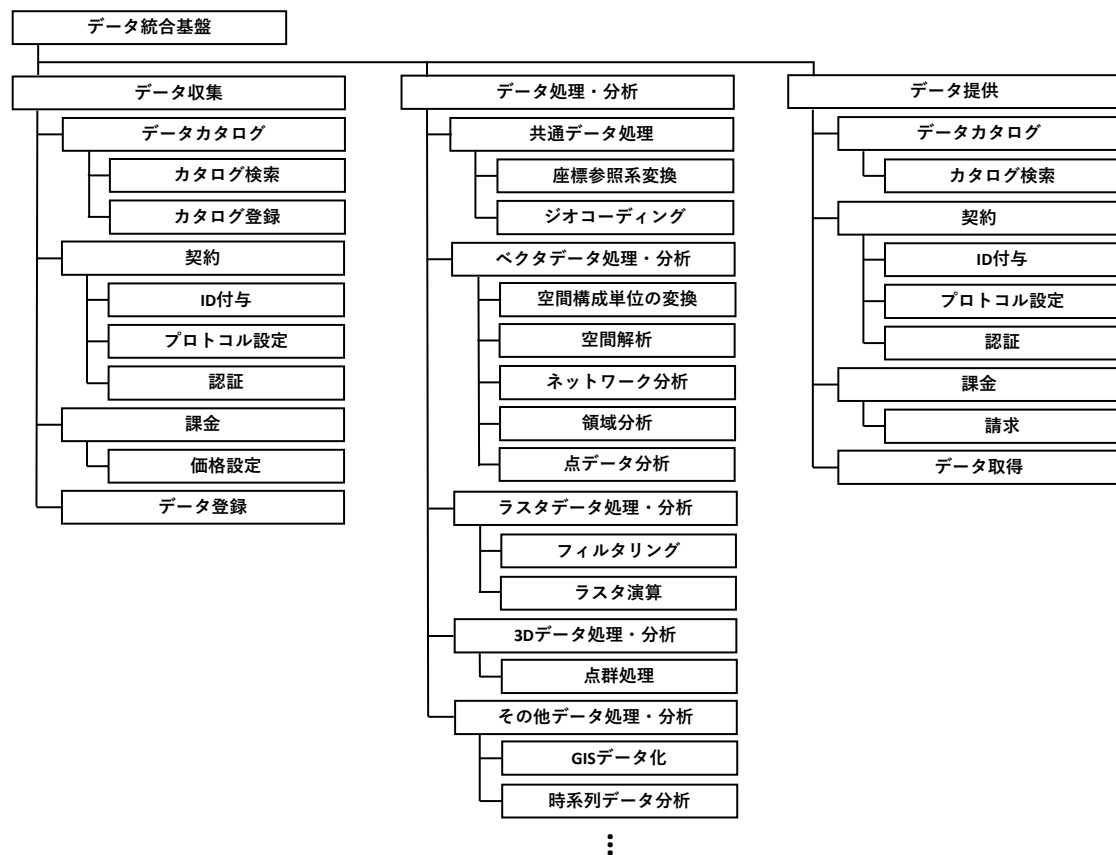


図 3.3: データ統合基盤の機能一覧

3.3.2 データモデル

本研究で提案するデータ統合基盤では、さまざまな種類の空間データを収集し、データ活用者やデータ加工者の要求に応じて適切にデータ提供を行うために管理する必要がある。これを実現する一つの方法はデータベース管理システム（DBMS : DataBase Management System）を用いることである。特に、地理情報科学の分野では、空間データベースシステムとして PostGIS（Postgresql の空間データ拡張）や SpatialLite（SQLite の空間データ拡張）などの RDBMS が開発されてきた。このような DBMS を用いてシステムを構築する場合、データモデルの設計が必要となる。

ここでは、データ統合基盤が扱うデータを管理、提供するための概念データモデル設計について述べる。

図 3.4、3.5 は、RDBMS における空間データのデータモデルを表している。

フィーチャ	
フィーチャID	
空間表現	
時間表現	
その他属性	

フィーチャID	空間表現	属性	...
0	 (ポリゴン)	施設A	
1	 (ポリゴン)	施設B	
⋮	⋮	⋮	

フィーチャID	空間表現	時間表現	属性	...
0	● (ポイント)	2022/01/15 09:00:00	人A	
1	● (ポイント)	2022/01/15 09:00:00	人B	
⋮	⋮	⋮	⋮	

図 3.4: 空間データモデル

図 3.5: 空間データのテーブルイメージ

フィーチャID は登録時にフィーチャ毎に自動で生成されることとする。空間表現は、座標群のことを指し、空間データの本体とも言える。時間表現は、動的な空間データの持つ時間情報に関する属性である。動的な空間データモデルには、デファクトスタンダードとして利用されているものがないため、さまざまなデータモデルが存在する。例えば、ArcGIS では、動的な空間データを表現するために、時間情報の属性として、特定の時刻を指す timestamp を用いるデータモデルと、特定の時刻範囲を指す start_at と end_at を用いるデータモデルの2種類が存在する。本研究では、前者の特定の時刻を指す属性として時間表現を用意する。この属性によって、例えば以下のような空間データの表現が可能である。

- 特定の時刻の人の位置情報
- 特定の時刻に利用されている公共施設情報
- 特定の時刻の台風の位置情報

3.3.3 「データ取得」API

本研究では、基盤のデータ提供の機能として「データ取得」APIを設計する。API (Application Program Interface) は、ソフトウェアコンポーネントの外部インターフェースのことを指すが、ここではHTTP プロトコルを利用して空間データを提供する WebAPI を想定している。

この API の役割は、「ユーザに指定された要件を満たすデータに対して、ユーザに指定されたデータ処理・分析を行った結果のデータをレスポンスデータとして提供する」ことである。

データ処理・分析手順の分析

「データ取得」API を設計するにあたり、ユーザがどのようなデータ処理・分析手順を実現したいかを検討する必要がある。

図 3.6 は想定されるデータ処理・分析手順を列挙したものである。今回、提案するデータ統合基盤の持つデータ処理・分析機能は一つまたは二つの入力に対して一つのデータを出力するものがある。そのため、一つの入力を要するデータ処理・分析と二つの入力を要するデータ処理・分析の両方に対応できるように API を設計する必要がある（図 3.6 の 2・3 行目）。また、一般的に複数のデータ処理・分析を行う必要がある場合も多くあると考えられるため、そのような複雑な手順にも対応できることが望ましい（図 3.6 の 4 行目）。

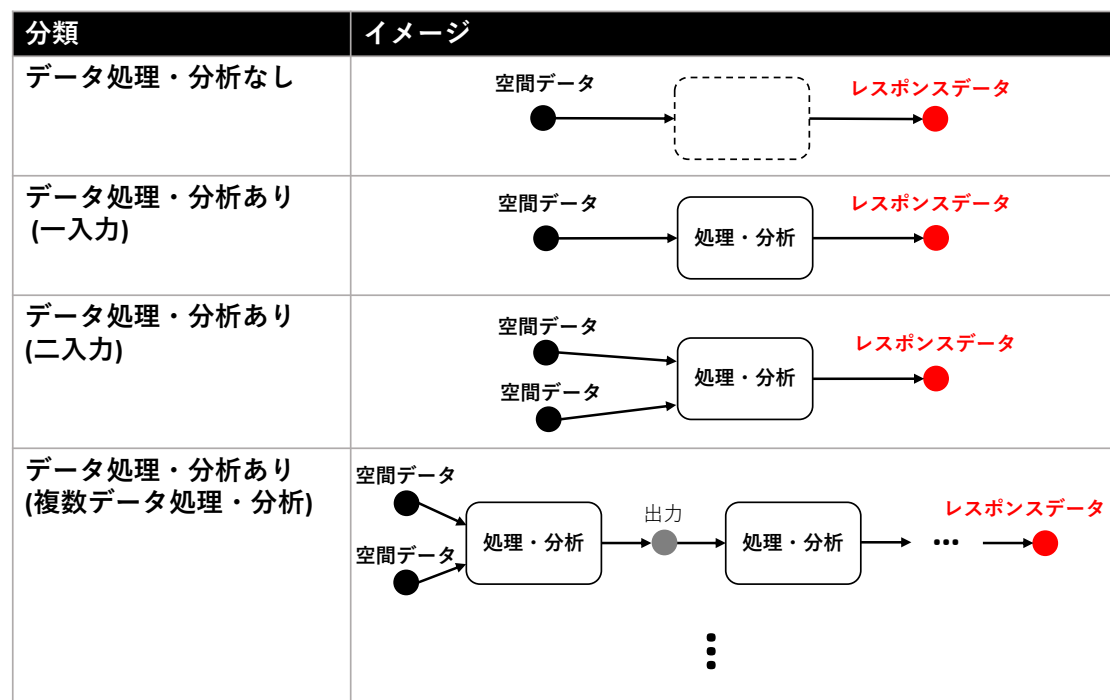


図 3.6: データ処理・分析手順の分類

エンドポイントの設計

- エンドポイント：http://api.example.com/{version}/data
- メソッド：GET

一般的な API と同様，API のバージョンを指す変数として version を用いる．また，データを取得することを明確にするために，パスに data を用いる．

さらに，この API は表 3.1 のクエリパラメータを用いる．

表 3.1: クエリパラメータ

クエリパラメータ	内容
プロセス	以下のパラメータで構成される． • データ名 • 検索条件 • データ処理・分析名 • データ処理・分析のパラメータ
データ名	データ処理・分析に対する入力データを指定．
検索条件 (任意)	以下の項目で入力データを絞り込む． • 空間範囲条件 • 時間範囲条件 • 属性条件
データ処理・分析名 (任意)	入力データに対して行うデータ処理・分析を指定．指定がない場合，入力データが直接提供される．
データ処理・分析 パラメータ (任意)	パラメータ設定が必要なデータ処理・分析に利用． (例：バッファによる領域分析の半径を指定)

空間範囲条件は，特定の空間範囲を指定することでデータを絞り込む．空間範囲条件の表現方法は表 3.2 の二通りを想定する（図 3.83.7）．

表 3.2: 空間範囲条件の表現方法

表現方法	内容
bbox	最小緯度，最大緯度，最小経度，最大経度からなる四角形の空間範囲を表現する．
circle	特定の地点（緯度，経度）と，半径によって定まる円形の空間範囲を表現する．

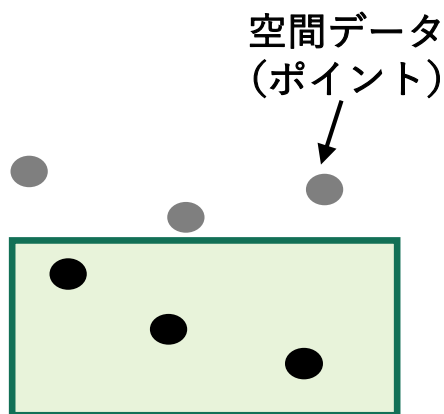


図 3.7: circle 型の空間検索

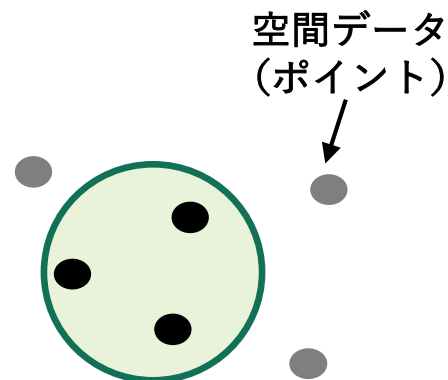


図 3.8: bbox 型の空間検索

時間範囲条件は、動的な空間データセットに対して、特定の時間範囲を指定することでデータを絞り込む。時間範囲条件の表現方法は表 3.3 のようになる。

表 3.3: 時間範囲条件の表現方法

表現方法	内容
start_at, end_at	開始時刻と終了時刻で時間範囲を表現する。
timestamp	特定の時刻で時間範囲を表現する。

属性条件は、特定の属性とその属性値を指定することでデータを絞り込む。属性条件の表現方法は、属性と属性値を直接指定する形になる。

これらの検索条件を指定することで、ユーザは、必要最小限のデータを抽出することが可能となり、結果としてアプリケーション側で保持するデータ量を削減することができるだけでなく、データ通信量を抑えることにもつながる。

レスポンスデータ

API のレスポンスデータを検討するにあたり、検討すべき点はデータ形式についてである。API のレスポンスデータのデータ形式は、Web クライアントのデフォルト言語である JavaScript との相性が高いという特徴から JSON(JavaScript Object Notation) 形式がデファクトスタンダードとなっている。

ベクタデータと 3D データの両モデルには、それぞれ GeoJSON[17] と CityJSON[18] といった JSON 形式のデータモデルが存在しているため、本研究で提案する API のレスポンスデータとしてはそれらの形式を採用する。ラスタデータに関しては、標準的な JSON 形式が存在しないため、JSON ベースのラスタデータの表現方法を設計する必要があるが、本研究では今後の課題とする。

第4章 実装

本章では，前章で提案した「データ取得」APIを実装する．

4.1 システム構築

図 4.1 は，本研究で開発したシステムの全体像を表している．

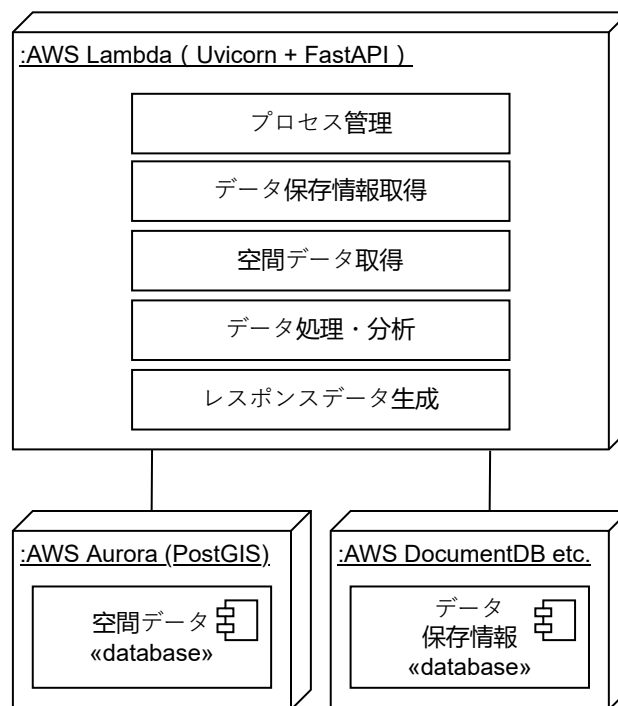


図 4.1: システムの配置図

空間データ

空間データベースとその管理には RDBMS である Postgresql の空間データ拡張である PostGIS[19] を使用する．PostGIS は，GIS ライブラリである GEOS の空

間演算機能を用いて高速な空間検索を行うことができる空間データベースである。格納される空間データの物理データモデルは図 4.2 の通りである。

空間表現には、geometry 型を用いる。このデータ型によって、地理座標系（緯度経度）で表現された空間データと投影座標系（xy 座標）で表現された空間データの両方を扱うことが可能となる。

時間表現には、timestamp 型を用いる。timestamp 型は日付と時刻の表記に関する国際規格である ISO 8601 に準拠したデータ型である。このデータ型によって、時間範囲条件による時間検索が可能となる。

その他属性部分はデータによってさまざまとなる。

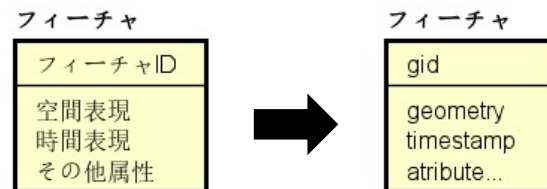


図 4.2: 概念データモデルと物理データモデルの対応

データ保存情報

PostGIS には、データ提供者から収集したデータに対して、データセット毎にテーブルを作成し保存する。この際、どのデータセットをどのテーブルに保存したかを表すデータ保存情報を管理する必要がある。この情報は MongoDB によって管理する。MongoDB は json 形式のデータを直接保存することが可能である NoSQL のデータベースである。データ保存情報のデータモデルは Listing 4.1 のようになる。

Listing 4.1: データ保存情報の例

```

1 {
2   "data_name": {data_name}, # データ名
3   "db_name": {db_name}, # データベース名
4   "table_name": {table_name}, # テーブル名
5   "data_category": {data_category_name} # データカテゴリ名
6 }
```

ここで、テーブル名だけでなくデータベース名とデータカテゴリ名を用意していることについて説明する。本研究の実装では、空間データベースとしてベクタデータを管理するデータベースを用いているが、このデータベースだけでは、ラスタデータや 3D データなどの他の分類の GIS データを管理することができない。

それらのデータはそのデータモデルの違いから、ベクタデータを管理するデータベースとは異なるデータモデルのデータベースを用意する必要がある。今後、それらのデータに対応するために別のデータベースを用意する場合、各データセットはどのテーブルに格納しているかだけでなく、どのデータベースに格納しているかという情報も必要になるため、データ保存情報にデータベース名を用意した。また、データカテゴリ名については、ベクタデータ、ラスタデータ、3D データの間でレスポンスデータの形式が異なるため、Web サーバがレスポンスデータを作成するに、データカテゴリ毎に適切な変換を行うために用意した。

これらの情報からなるデータ保存情報によって、Web サーバはユーザに指定されたデータ名のデータにアクセスし、レスポンスデータを作成することが可能となる。

Web サーバ

Web サーバは Python の Web フレームワークである FastAPI を用いて構築する。FastAPI は ASGI サーバである Uvicorn のアプリケーションとして HTTP リクエストに対する処理を行う。

この Web サーバの役割は、データ活用者またはデータ加工者からの「データ取得」の HTTP リクエストに対して、適切なデータ処理・分析プロセスを実行した結果となる空間データをレスポンスデータとして提供することである。詳細な実現方法は後述する。

4.2 「データ取得」 API

前章で示したように、「データ取得」API は「データ処理・分析なし」と「データ処理・分析あり」の二つの場合でクエリパラメータの構成が異なる。ここでは、それぞれの API について述べる。

「データ処理・分析なしのデータ取得」API

「データ処理・分析」を行わずにデータを取得する場合、ユーザが HTTP リクエストで指定する必要がある項目は、取得したい「データ名」と「空間範囲条件」「時間範囲条件」「属性条件」からなる「検索条件」である。これらの項目を `process` というクエリパラメータで設定する。

API は以下のような設計となる。

- メソッド：GET
- リソース：Listing4.2

Listing 4.2: HTTP リクエスト（データ処理・分析なし）

```
1 http(s)://api.example.com/{version}/data/?
2   process={
3     "data": [{data_name}], # データ名
4     "search_option": [
5       {
6         "spatial_range": {spatial_range}, # 空間範囲条件
7         "time_range": {time_range}, # 時間範囲条件
8         "properties": {properties} # 属性条件
9       }
10    ]
11  }
```

「データ処理・分析ありのデータ取得」API

「データ処理・分析」を含めてデータを取得する場合、ユーザがHTTP リクエストで指定する必要がある項目は、取得したい「データ名」と「空間範囲条件」「時間範囲条件」「属性条件」からなる「検索条件」に加えて、「データ処理・分析名」と必要に応じて「データ処理・分析に使用する変数」となる。これらの項目を前者と同様、process というクエリパラメータで設定する。また、process は、複数回指定することで、複雑なデータ処理・分析手順を行なった結果のデータを取得することが可能である。HTTP リクエストを受けとった Web サーバは、process の順に「データ処理・分析」を実行し、最後の process で得た結果をレスポンスデータとしてユーザに提供する。

API は以下のような設計となる。

- メソッド：GET
- リソース：Listing4.3, 4.4

Listing 4.3: HTTP リクエスト（データ処理・分析あり：1 入力）

```
1 http(s)://api.example.com/{version}/data/?
2   process={
3     "data_name": [{data_name}], # データ名
4     "search_option": [
```

```

5          {
6              "spatial_range": {spatial_range}, # 空間範囲条件
7              "time_range": {time_range}, # 時間範囲条件
8              "properties": {properties} # 属性条件
9          }
10     ],
11     "processing": {processing_name}, # データ処理・分析名
12     "processing_parameter": {val}, # データ処理・分析のパラ
        メータ
13 }

```

Listing 4.4: HTTP リクエスト（データ処理・分析あり：2 入力）

```

1 http(s)://api.example.com/{version}/data/?
2   process={
3       "data_name": [{data_name_1}, {data_name_2}], # データ名
4       "search_option": [
5           # {data_name_1} に対する検索条件
6           {
7               "spatial_range": {spatial_range}, # 空間範囲条件
8               "time_range": {time_range}, # 時間範囲条件
9               "properties": {properties} # 属性条件
10          },
11          # {data_name_2} に対する検索条件
12          {
13              "spatial_range": {spatial_range}, # 空間範囲条件
14              "time_range": {time_range}, # 時間範囲条件
15              "properties": {properties} # 属性条件
16          },
17      ],
18      "processing": {processing_name}, # データ処理・分析名
19      "processing_parameter": {val}, # データ処理・分析のパラ
        メータ
20 }

```

データ処理・分析

本研究では、第2章で示した空間データに対するデータ処理・分析の中でも共通データ処理・分析とベクタデータ処理・分析をデータ統合基盤の機能として実装する。ベクタデータは第2章でも示したように、基盤地図や国土数値、センサや人などさまざまな種類の空間情報を表現することが可能であり、さまざまなアプリケーション開発に必要な機能を担うことができると考えられる。

具体的なデータ処理・分析名とその内容の割り当ては表4.1の通りである。

表 4.1: データ処理・分析名の一覧

データ処理・分析名	内容	パラメータ
set_crs	座標参照系変換	"crs": EPSG コード
reverse_geocode	逆ジオコーディング	
aggregation	空間構成単位の変換：集約	"by": 属性名 "aggfunc": 集計方法 (first, sum, etc.)
simplify	空間構成単位の変換：単純化	"tolerance": 許容値 (m)
area	空間解析：基本量の測定 (ポリゴンの面積)	
length	空間解析：基本量の測定 (ラインの長さ)	
nearest	空間解析：オブジェクト選択 (最近隣データの選択)	"lng": 経度 "lat": 緯度
union	空間解析：オーバーレイ (union)	
intersect	空間解析：オーバーレイ (intersect)	
symmetric_difference	空間解析：オーバーレイ (symmetric difference)	
erase	空間解析：オーバーレイ (erase)	
identify	空間解析：オーバーレイ (identify)	
merge	空間解析：その他データ操作 (merge)	
clip	空間解析：その他データ操作 (clip)	
buffer	領域分析：バッファ	"length": 距離 (m)

「データ取得」の実現方法

Webサーバは「データ取得」のHTTPリクエストをユーザから受け取ると、図4.3の順に処理を行う。

Webサーバは、ユーザからのHTTPリクエストを受け取ると、まず、data_nameから、データ保存情報を取得する。その後、取得したデータ保存情報に含まれる、データベース名とテーブル名、そして検索条件であるsearch_optionに従い、空間データを取得する。取得された空間データは、適切なprocessing_nameの指定がある場合に、データ処理・分析を行う。ここまでの動作を「プロセス」のクエリパラメータの数分繰り返し、最終的に得られた空間データがレスポンスデータとしてユーザに提供される。

図4.3は、このデータ取得の一連の流れを示したものである。

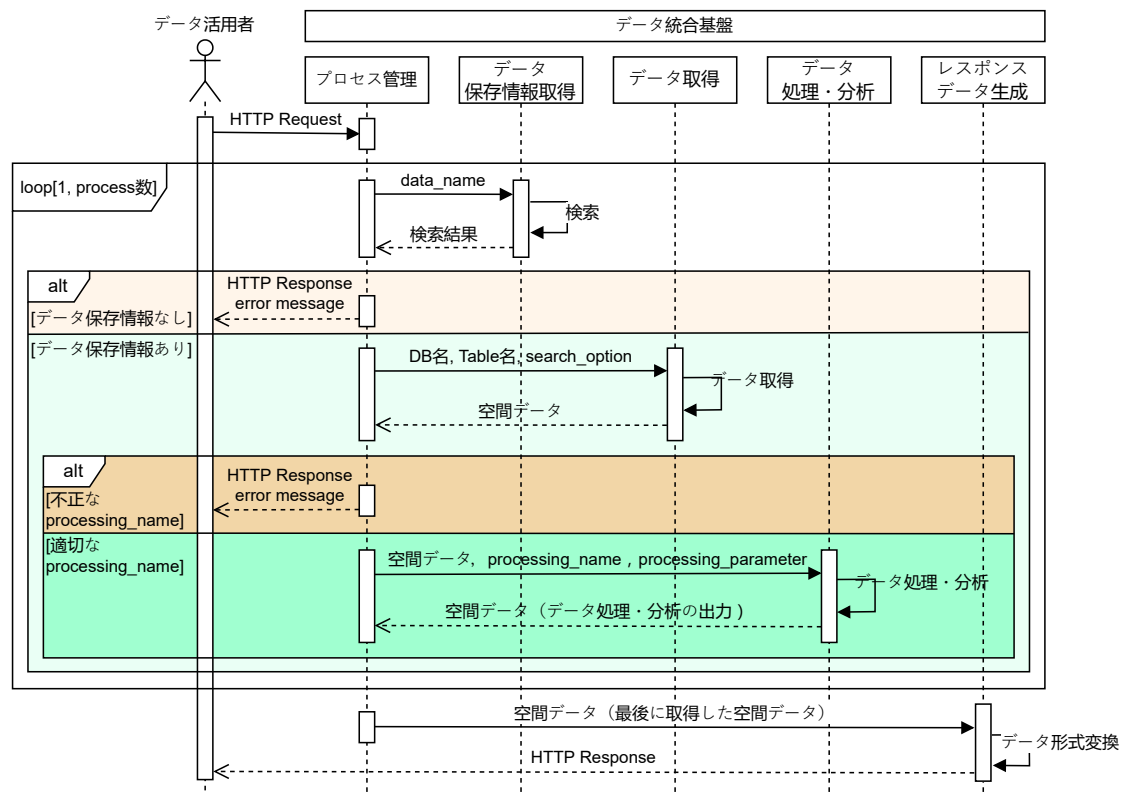


図 4.3: 「データ取得」のシーケンス図

レスポンスデータ

「データ取得」APIは、基盤上のベクタデータベースから取得したデータに対して、指定されたデータ処理・分析を実行した結果のデータをレスポンスデータ

としてユーザに提供する。

レスポンスデータのフォーマットには JSON 形式の空間データ拡張である GeoJSON 形式を採用する。レスポンスデータが JSON 形式であることで、ユーザはレスポンスデータを直接アプリケーションに活用することができる。

Listing4.5 は、ポイントデータの GeoJSON 形式の例である。

Listing 4.5: ポイントデータの GeoJSON 形式の例

```
1 {
2   "type": "FeatureCollection",
3   "crs": {
4     "type": "name",
5     "properties": {
6       "name": "urn:ogc:def:crs:OGC:1.3:CRS84"
7     }
8   },
9   "features": [
10    {
11      "type": "Feature",
12      "properties": { },
13      "geometry": {
14        "type": "Point",
15        "coordinates": [138.7309, 35.3628]
16      }
17    },
18    {
19      "type": "Feature",
20      "properties": { },
21      "geometry": {
22        "type": "Point",
23        "coordinates": [138.8079, 35.1983]
24      }
25    }
26  ]
27 }
```

第5章 実験

5.1 実験目的と方法

本研究で提案するデータ統合基盤は、データ活用者側でデータと処理・分析機能に要するリソースを削減すること、開発ステップを削減することを目的としている。また、このデータ統合基盤は複数種類のデータの収集、提供を実現するため、さまざまな分野のアプリケーションに必要なデータを提供することを目的とする。本節では、具体的なシナリオとして二種類のアプリケーションを検討し、アプリケーション実現のために必要なデータをデータ統合基盤の「データ取得」APIを用いて作成、取得できることを確認する。また、データ統合基盤の機能を用いてデータを取得することで、データと処理・分析機能に要するリソース、処理・分析機能の実装に要する開発ステップ数が削減されることを確認する。

5.2 実験環境

本研究で使用した実験環境は表 5.1, 5.2 の通りである。また、HTTP リクエストは API クライアント用のツールである Postman を用いて送信する。

表 5.1: 実験環境

CPU	Apple M1 chip
メモリ	16 GB
OS	macOS Big Sur version 11.6
開発言語	Python 3.8.11

表 5.2: Python ライブラリとデータベース

FastAPI (Web サーバ)	version 0.68.1
GeoPandas (データ処理・分析)	version 0.10.0
PostGIS (空間データ用 DB)	version 3.1
MongoDB (データ保存情報用 DB)	version 5.0.4

5.3 シナリオを用いた動作確認・比較評価

5.3.1 シナリオ 1：バス待ち人数の把握

シナリオの概要

まちづくりにおいて、自動車を利用できない高齢者等の交通弱者のために公共交通としてのバスを運営することは重要である [20]。利用者の要望に応じてバスの運行経路や運行時刻を決定するデマンド型交通のような仕組みを導入するために、リアルタイムの「バス待ち人数」を把握することは一つの解決策になると考えられる。

そこで、本研究では、石川県加賀市の加賀温泉駅のバス停にて、センサ等によって作成された人の位置情報からバス待ち人数データを作成するシナリオを検討する。このシナリオでは、表 5.3 のデータが基盤に登録されていることを前提とする。人の位置情報に関するデータは人流に関するオープンデータ等を参考に事前に表のようなデータを作成した。

表 5.3: 前提のデータ

データ (分類)	データ説明
人の位置情報 (ポイント)	・加賀温泉駅前（バス停付近を含む）の人の位置情報 ・2021-01-15 08:00:00 ～ （5 秒間隔で更新） ・各位置情報には人の ID が付与されている
バス停留所 (ポイント)	・石川県のバス停留所の位置情報 ・出典：国土交通省国土数値情報ダウンロードサイト (https://nlftp.mlit.go.jp/ksj/gml/datalist/KsjTmplt-P11.html)

アプリケーション側で活用したいデータは以下とする。

- バス停「加賀温泉駅」の 現在時刻（2021-01-15 09:0:10 を想定） のバス待ち人数を把握する

データ作成方法

バス待ち人数を把握するためには、人の位置情報の中から、「バス停領域内に含まれない人」と「バス停領域内には含まれるが、通過しているだけの人」を識別し、人数から除く必要がある。この作業を実現するための一つの方法として、以下の手順が考えられる。

1. バス停領域を決定するために、国土数値：バス停留所データに対して「領域分析：バッファ」を行う（今回は、バス停から半径 20m 以内をバス停領域とみなすことにする）。
2. バス停領域内の人のみを抽出するために人の位置情報に対してバス停領域を用いて「空間解析：データ操作（clip）」を行う（この際、人の位置情報は { "start_at": "2021-01-15 09:00:00", "end_at": "2021-01-15 09:00:00" } の時間範囲条件付きで取得する）。
3. 2 の出力となる人の位置情報に対して、時系列で各人の移動距離を計算し、閾値以下の移動距離の人をカウントする。

図 5.1 は、このデータ作成手順の地図上でのイメージを表している。

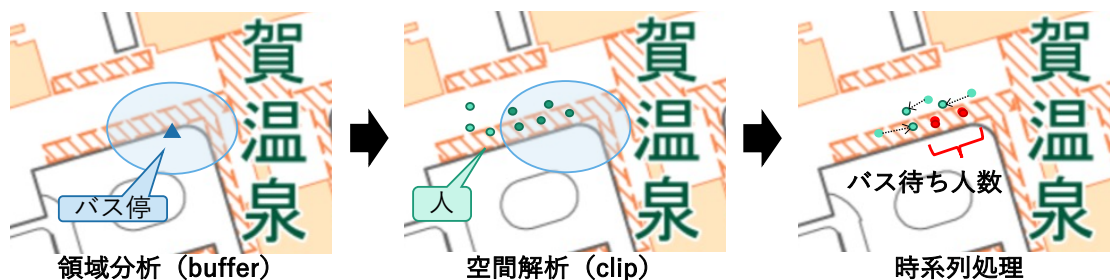


図 5.1: データ処理・分析手順のイメージ（出典：国土地理院地図を加工して作成）

また、図 5.2 は、このデータ作成手順の概念図である。データ統合基盤は、「領域分析：バッファ」と「空間解析：データ操作（clip）」のデータ処理・分析機能を要しているため、これらの処理を API のクエリパラメータによって指定することで、データ作成手順 2 の出力をユーザに提供することができる。

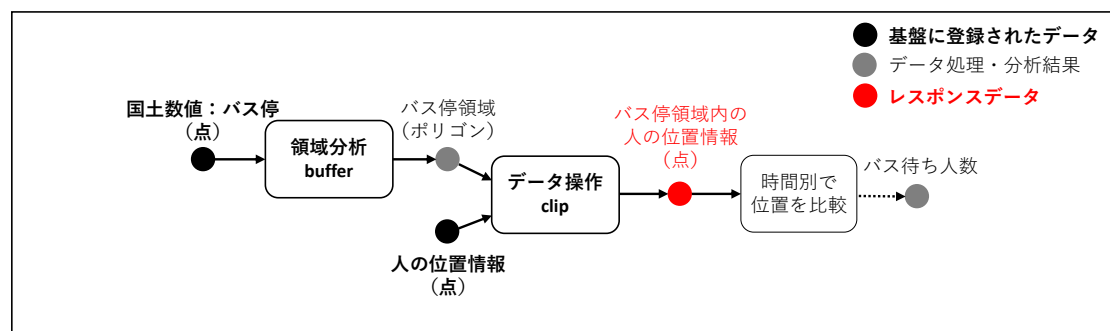


図 5.2: シナリオ 1 のデータ処理・分析手順

実行する HTTP リクエストは Listing5.1 のようになる。ここで, "p11_001": "加賀温泉駅" の部分は, バス停留所の名称の属性情報で抽出を行うためのパラメータである。

Listing 5.1: シナリオ 1 の HTTP リクエスト

```
1 http://localhost:8000/0/data/?
2   process={
3     "data_name": ["busstop"],
4     "search_option": [
5       {
6         "properties": {
7           "p11_001": " 加賀温泉駅"
8         }
9       }
10    ],
11    "processing_name": "buffer",
12    "processing_parameter": {
13      "buffer_area": 20
14    }
15  }&
16  process={
17    "data_name": ["busstop_people",1],
18    "search_option": [
19      {
20        "time_range": {
21          "start_at": "2021-01-15 09:00:00",
22          "end_at": "2021-01-15 09:00:10"
23        }
24      }
25    ],
26    "processing_name": "clip"
27  }
```

結果

データ統合基盤を介してデータを取得する場合（提案）と, すべての手順をアプリケーション側で行いデータを取得する場合（従来）を比較したものが表 5.4 である。この表から, このシナリオにおいてデータ統合基盤の機能を用いることで, 4 手順を 1 つの API で実行することができ, アプリケーション側が保持する必要があるデータ量を 99.8%削減できることがわかる。

表 5.4: 結果

	データ収集・処理・分析手順	アプリケーション側が保持するデータ量 (GeoJSON 形式)
従来	・データ収集：人の位置情報 ・データ収集：バス停 ・領域分析 (buffer) ・データ操作 (clip) ・時系列処理 (合計：5)	・人の位置情報 (0.9MB) ・バス停 (3.5MB) (合計：4.4MB)
提案	・API の利用 ・時系列処理 (合計：2)	・特定時間内、バス停領域内の人の位置情報 (0.007MB) (合計：0.007MB)

5.3.2 シナリオ 2：ドローン飛行可能空域の把握

シナリオ概要

ドローンの技術の発展に伴い、医薬品や食料、緊急時の物資をドローンによって配送する試みが多く、多くの地域で検討されている [21, 22]。この際、人口の多い住宅地などを避け、農地や森林上空をドローンの飛行経路とすることは、サービスによる事故の可能性を下げるために重要であると考えられる。

そこで、本研究では、石川県加賀市において配送用ドローンの飛行経路を決定する際に使用するデータとして、ドローン飛行可能空域を作成するシナリオを検討する。このシナリオでは、表 5.5 のデータが基盤に登録されていることを前提とする。静的飛行制限空域データは、加賀市内における空港付近、人口密集地域を示すポリゴンデータを事前に作成した。また、動的な飛行制限空域として、加賀市営のグラウンド等を参考に、公共施設のポイントデータを作成した。このデータは、30 分毎の施設の予約状況をもとに、timestamp が設定されたデータとする。

アプリケーション側で活用したいデータは以下とする。

- 2021-01-15 09:00:00～2021-01-15 09:30:00 のドローン飛行可能空域

データ作成方法

ドローン飛行可能空域を把握する方法として、以下の手順が考えられる。

1. 国土数値：森林地域から国土数値：行政区域のデータを用いて「空間解析：データ操作 (clip)」を行い、加賀市の森林地域を抽出する。
2. 1 の出力と加賀市の農地区画：筆ポリゴンのデータを「空間解析：データ操作 (merge)」によって結合する。

表 5.5: 前提のデータ

データ (分類)	データ説明
森林地域 (ポリゴン)	・ 石川県の森林地域のデータ ・ 出典：国土交通省国土数値情報ダウンロードサイト (https://nlftp.mlit.go.jp/ksj/gml/datalist/KsjTmplt-A13.html)
行政区域 (ポリゴン)	・ 石川県加賀市の行政区域のデータ ・ 出典：国土交通省国土数値情報ダウンロードサイト (https://nlftp.mlit.go.jp/ksj/gml/datalist/KsjTmplt-N03-v2_3.html)
筆ポリゴン (ポリゴン)	・ 石川県加賀市の農地区画のデータ ・ 出典：農林水産省・筆ポリゴンデータ (https://www.maff.go.jp/j/tokei/polygon/hudeporidl.html)
静的飛行 制限区域 (ポリゴン)	・ 空港周辺と人口密集区域のデータ
公共施設 利用状況 (ポイント)	・ 屋外施設の位置と利用状況のデータ ・ 2021-01-15 1 日分の利用状況 (30 分間隔)

3. 2 の出力から静的飛行制限空域のデータを「空間解析：オーバーレイ (erase)」によって除く。
 4. 利用中の公共施設の ({"start_at": "2021-01-15 09:00:00", "end_at": "2021-01-15 09:30:00"} の時間範囲条件付きで取得した) データに対して「領域分析：バッファ」を行うことで、動的飛行制限区域を作成する。
 5. 3 の出力から 4 の出力を「空間解析：オーバーレイ (erase)」によって除く。
- 図 5.3 は、このデータ作成手順の地図上でのイメージを表している。



図 5.3: データ処理・分析手順のイメージ (出典：国土地理院地図を加工して作成)

図 5.4 は、このデータ作成手順の概念図である。データ統合基盤は、上記で述べた全てのデータ処理・分析機能を要しているため、これらの処理を API のクエリ

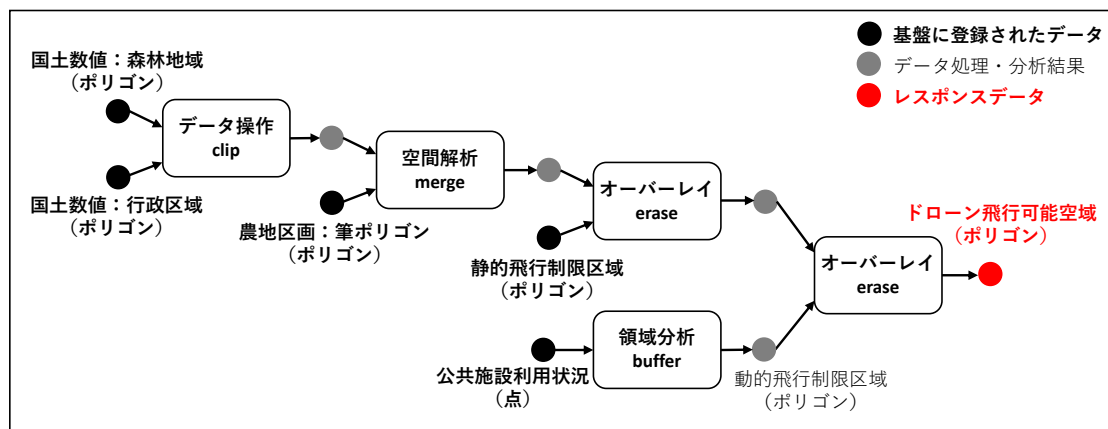


図 5.4: シナリオ 2 のデータ処理・分析手順

パラメータによって指定することで、データ作成手順 5 の出力をユーザに提供することができる。

実行する HTTP リクエストは Listing 5.2 のようになる。

Listing 5.2: シナリオ 2 の HTTP リクエスト

```

1 http://localhost:8000/0/data/?
2   process={
3     "data_name": ["forest","kaga_area"],
4     "processing_name": "clip"
5   }&
6   process={
7     "data_name": [1, "fude"],
8     "processing_name": "merge"
9   }&
10  process={
11    "data_name": [2,"static_drone_flightless_area"],
12    "processing_name": "erase"
13  }&
14  process={
15    "data_name": ["facility"],
16    "search_option": [
17      {
18        "time_range": {
19          "start_at": "2021-01-15 09:00:00",
20          "end_at": "2021-01-15 09:30:00"
21        }
22      }
23    ],
24    "processing_name": "buffer",
25    "processing_parameter": {
26      "length": 300

```

```

27     },
28   }&
29   process={
30     "data_name": [3,4],
31     "processing_name": "erase"
32   }

```

結果

データ統合基盤を介してデータを取得する場合（提案）と、すべての手順をアプリケーション側で行いデータを取得する場合（従来）を比較したものが表 5.6 である。この表から、このシナリオにおいてデータ統合基盤の機能を用いることで、10 手順を 1 つの API で実行することが可能となり、アプリケーション側が保持する必要があるデータ量を 75% 削減できることがわかる。

表 5.6: シナリオ 2 の結果

	データ収集・処理・分析手順	アプリケーション側が保持するデータ量（GeoJSON 形式）
従来	- データ収集：森林地域 - データ収集：行政区域 - データ収集：筆ポリゴン - データ収集：静的飛行制限空域 - データ収集：公共施設利用状況 - データ操作（clip） - データ操作（merge） - オーバーレイ（erase） - 領域分析（buffer） - オーバーレイ（erase） （合計：10）	- 森林地域（162.5MB） - 行政区域（0.34MB） - 筆ポリゴン（13.9MB） - 静的飛行制限空域（0.004MB） - 公共施設利用状況（0.013MB） （合計：176.8MB）
提案	API の利用 （合計：1）	ドローン飛行可能空域（44.2MB） （合計：44.2MB）

5.4 応答時間

API において、HTTP リクエストに対する応答時間は、機能外要求として重要な評価項目である。ここでは、本研究で提案した「データ取得」API の応答時間

について議論する。

5.4.1 シナリオの応答時間

計測結果

前述した二つのシナリオを 10 回行った際の応答時間は表 5.7 のようになる。一般的な API の応答時間が数十 ms から数百 ms なのに対して、シナリオ 2 では数百 s かかっていることがわかる。また、表 5.8 は、各データ処理・分析の計算時間を Web サーバ側で計測したものである。

表 5.7: シナリオの応答時間

	シナリオ 1	シナリオ 2
平均	122.1(ms)	149890.2(ms)

表 5.8: 各シナリオの各データ処理・分析の計算時間

データ処理・分析名		buffer	clip
平均計算時間 (ms)		13	5

データ処理・分析名	clip	merge	erase	buffer	erase
平均計算時間 (ms)	184154	6	5421	14	3239

考察

シナリオ 2 の応答時間が大きくなった理由としては、データサイズやデータ処理・分析の手順数などが原因として考えられる。特に、シナリオ 2 の最初の処理である「空間解析：clip」は入力となる空間データに「国土数値：森林地域」(162.5MB) という比較的大きなサイズのデータが含まれており、このデータに対する空間解析に計算時間を要していると考えられる。

一方で、シナリオ 2 に関しては、データ作成手順 3 の「空間解析：erase」までが、静的データを扱う処理であるため、事前にそこまでの処理を実行し、基盤に再登録することで、シナリオで実行するリクエストのデータ処理・分析手順を図 5.5 のように削減することができる。

前述したシナリオと同様に、この手順に対応する HTTP リクエストを 10 回実行した結果の平均応答時間は 7757.6(ms) となった。このように、提案するデータ統合基盤では、一度取得したデータを基盤に登録することで、そのデータを再度作成せずに活用することが可能となり、そのデータを活用しデータを作成する HTTP

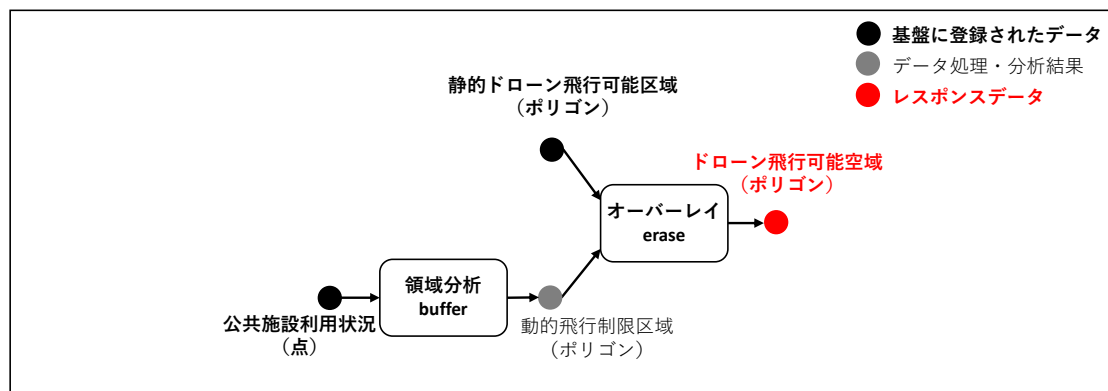


図 5.5: 一部を事前に実行済みのシナリオ 2 のデータ処理・分析手順

リクエストの応答時間を短縮することが可能となる。今回のシナリオでは、事前に静的ドローン飛行可能空域データを作成・登録しておくことで、応答時間を約 95%短縮することができた。

5.4.2 データサイズと応答時間

シナリオの応答時間で得られた考察より、データサイズによって応答時間がどのように変化するかを実験で確認する。

ここでは、表 5.9 のような異なるデータサイズの空間データを用意する（この表では GeoJSON 形式のデータサイズを示している）。ここで、各ポリゴンは緯度経度それぞれ 0.01 度幅の四角形のポリゴンとして作成した。

表 5.9: テストデータ

	1	2	3	4	5	6	7	8	9	10
ポリゴン数	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
サイズ (MB)	0.6	1.3	1.9	2.6	3.2	3.9	4.5	5.2	5.8	6.5

テスト内容

- API のクエリパラメータ：データ名のみ（検索条件、データ処理・分析なし）
- 回数：100

計測結果

図 5.6 は、データ取得の平均応答時間を示している。

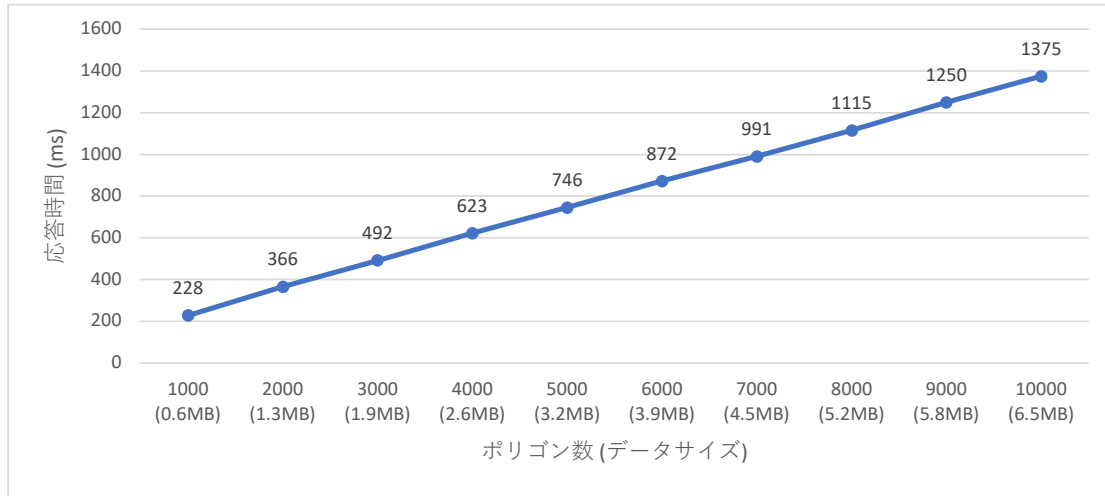


図 5.6: データ取得（処理・分析なし）の平均応答時間

考察

図 5.6 から、データ処理・分析を介さずにデータを取得する場合、その応答時間は空間データのフィーチャ数（今回の場合ポリゴン数）に従い増加することが分かる。データ処理・分析を介する場合も、同様にフィーチャ数が多いほど応答時間が長くなると考えられる。

データ統合基盤が支援できるアプリケーションの分野は、このような応答時間を考慮して分析する必要がある。例えば、リアルタイム性を要するアプリケーションには、要求される応答時間や要求されるデータのサイズと、想定される応答時間を比較する必要がある。

第6章 議論

本章では，提案するデータ統合基盤の設計において，今後の展望として検討すべき点について述べる．

6.1 APIの設計

機能外要求

本研究で提案した「データ取得」APIは，前章のシナリオを用いた実験で示したように，データサイズやデータ処理・分析の種類や手順によって応答時間が大きく変化する．実際に，データ統合基盤を運用する場合は，応答時間以外の同時リクエスト数やCPU使用率などの機能外要求の項目についてまとめる必要がある．

また，それらの機能外要求に応えるために，必要に応じて「データ取得」APIで取得することができるデータのサイズやデータ処理・分析手順などのクエリパラメータに上限を設ける必要性などが考えられる．

このような計算機資源に関する機能外要求は，多くの分野でアプリケーションプラットフォームを実用化する際に生じる問題であるため，この問題を解決するためにクラウドを活用することが検討されている[26][27]．特に，Amazon Web Services (AWS)，Google Cloud Platform，Microsoft Azure のサービスは，ユーザを計算機資源に関するタスクから解放するための機能が多く含まれており，これらのクラウド技術を用いてデータ統合基盤を検討することも今後の方向性として考えられる．

レスポンスデータの設計

本研究の実装では，「データ取得」APIのレスポンスデータとして，GeoJSON形式の空間データをレスポンスデータとして提供したが，実際には，提供される空間データがどのようなデータを示すためのメタデータを付与する必要がある．以下では，具体的に加える必要があると考えられる項目について述べる．

- 実行したデータ処理・分析手順（データ名，データ処理・分析名など）

実行したデータ処理・分析手順）に関する情報は、アプリケーション側が想定したデータ処理・分析手順を実行できているかどうかを確認するために必要であると考えられる。

- 粒度

さまざまなデータ提供者から収集された空間データには、その作成方法や活用目的の違いから、空間粒度や時間粒度が異なるという特徴がある。このような粒度の違いは、アプリケーション側のサービス品質に影響を与えるため、レスポンスデータが実際にどのような粒度のデータであることを示すパラメータ等を用意する必要があると考えられる。空間データの粒度に関してのより詳しい議論は、地理情報科学の分野で不確実性（Uncertainty）という研究分野があるため、そちらを参考にすることが望ましい。

また、エラー処理についてもテストケースを網羅的に用意し、動作確認を行う必要がある。

6.2 データ処理・分析手順の実現方法

本研究で提案した「データ取得」の実現方法は、図 6.1 の左図のような逐次型の方法となっている。一方でこのような構造のデータ処理・分析は、図 6.1 の右図のように、互いに依存しないデータ処理・分析を並列で行うことで、HTTP リクエストに対する応答時間を短縮できることが考えられる。そのためには、データ処理・分析手順を分析し、並列処理可能である部分を抽出する機能が必要となる。

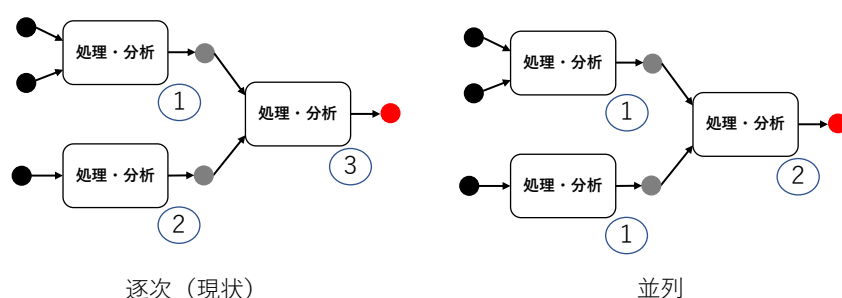


図 6.1: データ処理・分析手順の実現方法のイメージ

6.3 データベース

ラスタデータ・3D データ・動的な空間データ

本研究の実装では、ベクタデータに対応するデータベースとして PostGIS を用いた空間データベースを構築した。一方で、第2章で示したように、空間データにはラスタデータや3D データといった異なるデータモデルのものが存在する。これらのデータをベクタデータと同様、データベースによって管理し、ユーザに提供するためには、それぞれのデータモデルに対応した別のデータベースを用意する必要がある。

ラスタデータに関しては、PostGIS にラスタデータのデータモデルに対応するための拡張機能が存在するため、それらを利用することが一つの手段として考えられる。

3D データに関しては、3D 都市モデルの代表的なデータ形式である CityGML に対応するために開発された 3DCityDB[23] が存在する。このデータベースは PostGIS にも対応しており、提案するデータ統合基盤の既存の RDBMS でのデータの管理が可能であると考えられる。

また、動的な空間データの表現方法もさまざまな種類が存在する。本研究ではデータモデルに時間表現を表す timestamp を用いることで、動的な空間データを表現したが、そのような時間表現が付与された空間データベースとして MobilityDB[24] が提案されている。このデータベースも PostGIS に対応しているため、提案するデータ統合基盤の既存の RDBMS で管理することが可能であると考えられる。また、移動体を表現する空間データモデルには、mf-json と呼ばれる OGC 標準も提案されており [8]、このようなデータモデルがデータ統合基盤に有効であるかどうかは検討すべきであると考えられる。

SQL vs NoSQL

本研究では、空間データのデータベースとして RDBMS である PostGIS を採用した。PostGIS は、GIS ライブラリを利用して高速な空間検索が可能であるという特徴がある。

- ベクタやラスタなどのさまざまな種類のデータモデルに対応できる
- 高速な空間検索が可能（R 木による空間インデックス）

一方で、NoSQL を空間データベースとして利用することも検討されており、Guo ら [25] は NoSQL の利点として以下のような点があると述べている。

- 多数の同時ユーザーリクエストをサポートしながら、任意の大規模な空間データを保存、管理できる
- データベース内のレコード数が多い場合、ドキュメントデータベースの方がSQL データベースよりも応答時間や行間の問い合わせが速い

現在は、空間データに対する機能に関しては、PostGIS を代表としてSQL が豊富な機能を有しているが、今後、NoSQL において空間データに対するデータ処理・分析機能の開発が進められれば、提案するデータ統合基盤のデータベースとしてNoSQL が有利となることも考えられる。

第7章 おわりに

結論

本研究では、空間情報活用のためのデータ統合基盤として、アプリケーションに必要なデータを取得するためのAPIを提案した。

空間情報には、さまざまな種類が存在し、それらのデータを組み合わせてアプリケーションを実現する場合、開発者は、必要なデータを収集し、適切なデータ処理・分析手順を実現することで、アプリケーションに必要な空間情報を導出する必要があった。

提案した「データ取得」APIは、検索条件として、空間範囲、時間範囲、属性を指定することで、アプリケーションの実現のための必要最低限のデータを絞り込んで取得することが可能となる。また、ユーザは必要に応じて、データ処理・分析機能を指定することが可能であり、特定のデータに対してアプリケーションの実現のために必要なデータ処理・分析を実行した結果となるデータを取得することが可能となる。また、一度ユーザが基盤のデータ処理・分析機能を用いて作成、取得したデータは、再度データ統合基盤に登録することで、他のユーザがデータ処理・分析手順を考慮せずに、データを容易に取得することが可能となる。

実験では、具体的に二つのシナリオに対してAPIを適用し、ユーザが要求するデータ処理・分析を実行したデータを取得できることが確認できた。これにより、各シナリオに対して、アプリケーション側でデータ収集、データ処理・分析に要する手順をそれぞれ60%, 90%, アプリケーション側で保持するデータ量をそれぞれ99.8%, 75%削減することができた。

今後の展望

提案するデータ統合基盤の価値を示すためには、空間情報を活用するアプリケーションのシナリオをより具体的に網羅し、どの程度のシナリオに対して基盤の機能を利用できるかを定量的に評価する必要がある。

また、このデータ統合基盤は、複数ユーザに同時に利用されることを想定しているため、応答時間、同時リクエスト数などの機能外要求に関して網羅的に負荷

テストを行い、設計についての評価を行う必要がある。応答時間の短縮については、データ処理・分析手順の並列化が一つの手段と考えられる。

最後に、本研究ではベクタデータに対応した空間データベースとして PostGIS を用いたが、ラスタデータや 3D データ、移動体データといった他のデータモデルに柔軟に対応するために、PostGIS のラスタ拡張や 3DCityDB, MobilityDB といった別のデータベースを構築することが考えられる。これにより、より幅広いアプリケーションのシナリオに対応することができると考えられる。

謝辞

本論文を執筆するにあたり，多くの方からご助言とご助力を頂きました．主指導教員の丹康雄教授には，日頃のゼミで本研究の議論を始め，対外発表に向けた論文の執筆や発表資料作成の際に熱心で丁寧なご指導を賜りました．また，本研究に留まらず研究室での業務などを通じて情報通信分野全般にわたり知見を深める機会をいただきました．心より感謝申し上げます．審査委員を引き受けてくださったリム勇仁准教授，篠田陽一教授，鈴木正人准教授，には，新たな視点から示唆に富んだアドバイスをいただきました．リム勇仁准教授には副指導教員を引き受けていただき，日頃のゼミを通じて活発な議論，熱心なご指導を賜りました．篠田陽一教授には，中間発表等の機会において，研究の方向性に関して重要なご意見をいただきました．鈴木先生にはソフトウェア工学の視点から多くの知見をご教示いただきました．また，丹・リム研究室の皆様には，論文や発表資料の添削などの研究活動ばかりでなく日常生活においてもご協力いただきました．厚く感謝申し上げます．

最後に，学生生活を支えてくださった家族に感謝いたします．ありがとうございました．

本研究に関する对外発表

- 亀田晃佑・辛涛・リム勇仁・丹康雄, ”空間情報活用のためのデータ統合基盤に関する研究”, 第 72 回 UBI 研究会, Nov. 2021.
- 亀田晃佑・辛涛・リム勇仁・丹康雄, ”空間情報活用のためのデータ統合基盤に関する研究”, 情報処理学会第 84 回全国大会, Mar. 2022.

参考文献

- [1] 関本義秀. リアルタイム化する地理空間情報ビジネス, 写真測量とリモートセンシング, 一般社団法人日本写真測量学会, vol.54, no.4, pp.211–215 (2015)
- [2] Wagemann, J., Clements, O., Marco, F. R., Rossi, A. P., and Mantovani, S. Geospatial web services pave new ways for server-based on-demand access and processing of Big Earth Data, *International Journal of Digital Earth*, vol.11, no.1, pp.7–25 (2018)
- [3] Döllner, J., Baumann, K., and Buchholz, H. Virtual 3D city models as foundation of complex urban information spaces, *Proc. 11th international conference on Urban Planning and Spatial Development in the Information Society*, pp.107–112 (2006)
- [4] Kolbe, T. H., Gröger, G., and Plümer, L. CityGML: Interoperable access to 3D city models, *Geo-information for disaster management*, pp.883–899 (2005)
- [5] Penttilä, H. Describing the changes in architectural information technology to understand design complexity and free-form architectural expression, *Journal of Information Technology in Construction (ITcon)*, vol.11, no.29, pp.395–408 (2006)
- [6] Weinmann, M., Jutzi, B., and Mallet, C. Feature relevance assessment for the semantic interpretation of 3D point cloud data, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol.5, no.W2, pp.1 (2013)
- [7] 浅見泰司, 矢野桂司, 貞広幸雄, 湯田ミノリ (編著). 地理情報科学 - GIS スタンダード, 古今書院 (2015)
- [8] Asahara, A., Hayashi, H., and Ishimaru, N., Shibasaki, R., and Kanasugi, H. International Standard “OGC Moving Features” to address “4Vs” on locational BigData, 2015 IEEE International Conference on Big Data (Big Data) pp.1958–1966 (2015)

- [9] Steiniger, S., Weibel, R., and Warf, B. GIS software: a description in 1000 words, Sage, (2010)
- [10] Zhao, P., Foerster, T., and Yue, P. The geoprocessing web, *Computers & Geosciences*, vol.47, pp.3–12 (2012)
- [11] MapServer, <https://download.osgeo.org/mapserver/docs/MapServer.pdf> (参照日 2022-1-18)
- [12] Zhao, P., Di, L., Han, W., and Li, X. Building a web-services based geospatial online analysis system, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol.5, no.6, pp.1780–1792 (2012)
- [13] Wei, Y., Santhana-Vannan, S. K., and Cook, R. B., Discover, visualize, and deliver geospatial data through OGC standards-based WebGIS system, 2009 17th International Conference on Geoinformatics, pp.1–6 (2009)
- [14] Wei, Y., Liu, S., Huntzinger, D. N., Michalak, A. M., Viovy, N., Post, W. M., Schwalm, C. R., Schaefer, K., Jacobson, A. R., Lu, C., and others, The North American carbon program multi-scale synthesis and terrestrial model inter-comparison project–Part 2: Environmental driver data, *Geoscientific Model Development*, vol.7, no.6, pp.2875–2893 (2014)
- [15] Steiniger, S., and Hunter, A.J.S. Free and open source GIS software for building a spatial data infrastructure, *Geospatial free and open source software in the 21st century*, pp.247–261 (2012)
- [16] Rajabifard, A., and Williamson, I. P. Spatial data infrastructures: concept, SDI hierarchy and future directions, *Geomatics '80*, Tehran, Iran (2001)
- [17] Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., Schaub, T., and others. The geojson format, Internet Engineering Task Force (IETF), (2016)
- [18] Ledoux, H., Ohori, K. A., Kumar, K., Dukai, B., Labetski, A., and Vitalis, S. CityJSON: A compact and easy-to-use encoding of the CityGML data model, *Open Geospatial Data, Software and Standards*, vol.4, no.1, pp.1–12 (2019)
- [19] Ramsey, P., and Columbia, V.B. Introduction to postgis, *Refractions Research Inc*, pp.34–35 (2005)

- [20] 長谷川大輔, 鈴木勉. 路線網形状を考慮したバス・デマンド型交通併用効果の分析, GIS-理論と応用, vol.27, no.1, pp.1–11 (2019)
- [21] Scott, J., and Scott, C. Drone delivery models for healthcare, Proceedings of the 50th Hawaii international conference on system sciences (2017)
- [22] Frachtenberg, E. Practical drone delivery, Computer, vol.52, no.12, pp.53–57 (2019)
- [23] Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubauer, A., Adolphi, T., and Kolbe, T. H. 3DCityDB-a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML, Open Geospatial Data, Software and Standards, vol.3, no.1, pp.1–26 (2018)
- [24] Zimányi, E., Sakr, M., and Lesuisse, A. MobilityDB: A mobility database based on PostgreSQL and PostGIS, ACM Transactions on Database Systems (TODS), vol.45, no.4, pp.1–42 (2020)
- [25] Guo, D., and Onstein, E. State-of-the-art geospatial information processing in NoSQL databases, ISPRS International Journal of Geo-Information, vol.9, no.5, pp.331 (2020)
- [26] Bebortta, S., Das, S. K., Kandpal, M., Barik, R. K., and Dubey, H. Geospatial serverless computing: Architectures, tools and future directions, ISPRS International Journal of Geo-Information, vol.9, no.5, pp.311 (2020)
- [27] Mete, M. O. and Yomralioglu, T. Implementation of serverless cloud GIS platform for land valuation, International Journal of Digital Earth, vol.14, no.7, pp.836-850 (2021)