

4次元時空間情報利活用の ための空間 ID ガイドライン (1.0 beta 版)

経済産業省

国土交通省

国土地理院

国立研究開発法人 新エネルギー・産業技術総合開発機構

独立行政法人 情報処理推進機構

2025年（令和7年）3月21日

改定履歴

改定年月日	改定箇所	改定内容
2023年04月28日	-	β 初版発行
2024年02月29日		γ 版発行 ・主な修正点 ・3.2.2 命名規則を修正 ・3.3.1 ドローン領域の API を修正 ・3.3.2 地下埋設物領域の空間 ID 紐付けを追記 ・3.3.4 自動車領域を新規追加
2025年3月21日		1.0 beta 版発行 ・主な修正点 ・ガイドライン名称の修正 ・1章 名称を修正 ・大幅な構成の見直し、記載の更新および修正

目次

1.	空間 ID による 4 次元時空間情報の利活用に係る取組の背景及び目的.....	3
1.1.	背景	3
1.2.	本ガイドラインの目的と構成.....	4
1.3.	用語の定義	4
2.	空間 ID の概要と定義.....	5
2.1.	空間 ID の概要.....	5
2.2.	空間 ID の特性と意義.....	5
2.3.	空間ボクセル.....	6
2.3.1.	空間ボクセルの概念.....	6
2.3.2.	空間ボクセルによる空間分割方式の基本要素.....	7
2.3.3.	空間ボクセルのズームレベルとサイズ.....	11
2.4.	空間 ID.....	14
2.4.1.	空間 ID の形式.....	14
2.4.2.	空間 ID の各インデックスの算出方法.....	14
2.4.3.	高さ情報を持たないデータに対応した空間 ID の形式.....	15
2.5.	拡張仕様	16
2.5.1.	ローカル空間 ID	16
2.5.2.	時空間 ID	24
3.	空間 ID の使用方法.....	27
3.1.	空間 ID の基本的な使い方.....	27
3.1.1.	識別子としての空間 ID	27
3.1.2.	空間ボクセルへの属性付与.....	27
3.1.3.	時間情報の扱い.....	29
3.1.4.	空間ボクセルに地物等の情報を紐付ける際のズームレベルの選択.....	30
3.2.	空間 ID 対応システム.....	31
3.2.1.	システム概要.....	31
3.2.2.	空間 ID 対応システムの基本構造.....	33
3.2.3.	API 設計方針.....	34
3.2.4.	空間 ID 変換に係る機能.....	42
3.2.5.	インポート機能.....	43
3.3.	ユースケース実証事例.....	50
3.3.1.	空間 ID の活用可能性.....	50
3.3.2.	ユースケース実証事例.....	51
3.4.	空間 ID 対応システムの運用.....	52
4.	今後の展望	53
4.1.	ロードマップ.....	53
4.2.	普及及び標準化に向けた取組.....	53

Appendix-1 用語集

Appendix-2 API 仕様例

Appendix-3 事例集

1. 空間 ID による 4 次元時空間情報の利活用に係る取組の背景及び目的

1.1. 背景

人とロボットやシステムが共生する Society5.0 の実現に向けては、仮想空間（サイバー空間）と現実世界（フィジカル空間）を高度に融合させる必要があり、フィジカル空間の空、地上、地下、屋内、海といったあらゆる空間をサイバー空間に双子のように再現した仕組み（デジタルツイン）を構築し、デジタルツイン上で行われるデータの連携や分析を、現実世界にフィードバックする仕組みが重要である。

しかし、空間情報は異なる仕様・規格で様々な主体によって特定の目的に特化して整備・管理され、共通的なメタ情報が付与されていないために、一元的な探索や集約・重畳が困難であり、空間情報を扱うシステムを構築する際の課題になっている。また、空間を 3 次元で一意に定義する共通規格が存在していないために、データ連携の際にはジオメトリの変換や情報の範囲・区切りを揃えるための変換コストが発生する。

そこで、異なる基準に基づいた空間情報であっても空間を一意的に識別可能な共通の識別子を用いて位置を特定するとともに、相互変換に用いる 3 次元の共通的な規格として、空間 ID¹ を導入し、様々な空間情報を時間情報も含めて効率的に処理するための技術開発・標準化を行う。



図 1-1 フィジカル空間のあらゆる空間をサイバー空間に再現のイメージ

¹ 経済産業省では、デジタル化をイネーブラーとして、企業と企業をつなぐビジネス・デジタルの協調領域を整備し、利用可能とすることにより、産業界全体として新たな連携による価値を生み出すエコシステムである「ウラノス・エコシステム」を推進している。ウラノス・エコシステムにおいて、仕様や規格が異なる空間情報（時間情報を含む）を簡単に統合・検索して、自由に流通させるための役割を果たすのが「ウラノス 4 次元時空間 ID」（通称：空間 ID）であり、3 次元空間における識別子「空間 ID」に時間軸における識別子「時間 ID」を追加したものを「時空間 ID」とする。

1.2. 本ガイドラインの目的と構成

(1) 目的

本ガイドラインは、空間 ID 及び空間 ID 対応システムについて、事業運営者から開発者まで幅広い読者に対して、ユースケースを例示しながら、技術仕様や運用指針を示すものである。本ガイドラインを通じて、空間 ID 及び空間 ID 対応システムが普及し、4次元時空間情報の利活用が広がっていくことを目的とする。

(2) 推進体制

本ガイドラインの策定にあたっては、独立行政法人情報処理推進機構のデジタルアーキテクチャ・デザインセンター（以下、「DADC」という。）を中心に官民で連携して取組を進めてきた。具体的には、経済産業省がデータ流通その他全般の観点から、国土交通省が3D都市モデルの活用・普及の観点から、国土地理院が空間 ID の定義の観点から、DADC はアーキテクチャ設計の観点から、国立研究開発法人新エネルギー・産業技術総合開発機構（以下、「NEDO」という。）には開発・実証の観点から、その取組を推進してきた。

(3) 構成

本ガイドラインでは、1章に取組の背景・目的として実現したい社会の仕組みと設計方針、想定読者を示す。2章に空間 ID の概要と定義および活用可能性、空間ボクセルの概念と空間分割方式・算出方法を示す。3章に空間 ID の基本的な使用方法・システム概要と実証事例、運用方針を示す。4章に今後の展望として社会実装のロードマップ、普及及び標準化に向けた取組と考え方を示す。

(4) 想定読者

本ガイドラインの主たる読者は、空間属性情報を活用した事業を運営するデータ活用事業運営者、データ活用事業運営者が用いるアプリを開発するアプリ開発者、空間 ID 対応システムの運営者・開発者、空間属性情報を提供するデータ提供者、これらのデータを収集・提供・受信する機器を提供する周辺機器事業者の5者を想定している。

1.3. 用語の定義

本ガイドラインで使用する用語の定義は APPENDIX-1 の用語集に示す。

2. 空間 ID の概要と定義

2.1. 空間 ID の概要

地球上の特定の空間領域を一意に識別するための識別子が空間 ID である。空間領域の単位は、3 次元空間を直方格子状に分割した直方体（以下、「空間ボクセル」という。）とする。空間ボクセルを再帰的に分割することにより、地球規模から数十センチメートル規模に至るまでの様々なサイズの空間ボクセルを定義できる。個々の空間ボクセルには、グローバルに一意の識別子である空間 ID を割り当てる。

各空間ボクセル内に存在する地物等の情報を空間 ID と紐付けて、空間 ID を共通の識別子として使用し、各情報を配信する際の空間的な単位を空間ボクセルとすることにより、統一された基準による情報流通が可能となり、各情報の検索や統合が容易になる。また、各情報を空間ボクセル単位で抽象化することで空間の概況を把握することもできる。

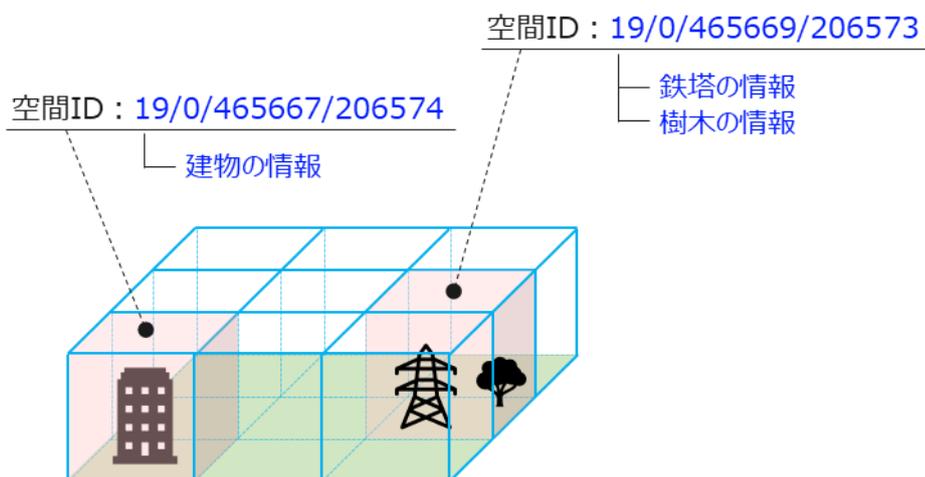


図 2-1 空間 ID のイメージ

2.2. 空間 ID の特性と意義

空間 ID は、4 次元時空間を扱う、ID 採番ルールを共通化する、計算により ID を算出できる、階層性を持つ ID 体系といった特性がある。また、空間 ID を活用することで、相互接続性を確保できる、空間情報の効率的・柔軟な活用ができる、情報処理及び通信負荷を削減できる、という意義がある。具体的な活用メリットはこれらの特性と意義を元に各ユースケースに応じて具体的に生まれる。図 2-2 に空間 ID の特性と意義およびそれぞれの項目の説明と関係性を示す。

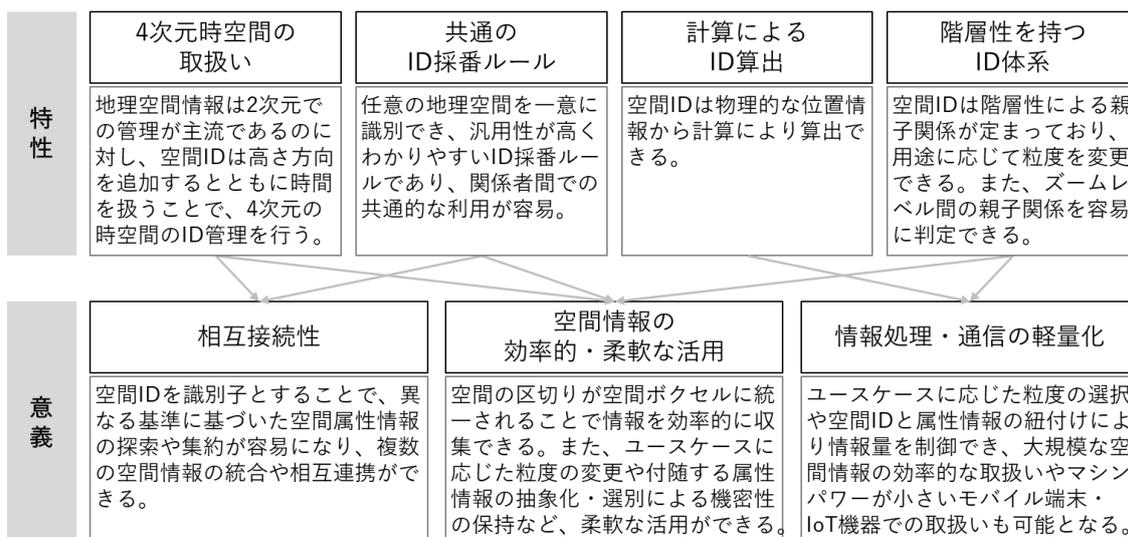


図 2-2 空間 ID の特性と意義

2.3. 空間ボクセル

2.3.1. 空間ボクセルの概念

空、地上、地下、屋内、海を含む地球上のあらゆる空間を直方格子状に分割した際の個々の直方体の空間領域を「空間ボクセル」とする。各空間ボクセルには、一意に位置を特定できる「空間 ID」を付与する。

空間ボクセルは以下のような構造を持つ。

- ・ 最上位の階層をズームレベル0とし、ズームレベルが1つ増えるごとに空間ボクセルの8分割を繰り返す階層構造を持つ。
- ・ ズームレベル間で親子関係を持つ。
- ・ 同一ズームレベルにおいて重複する空間ボクセルは存在しない。

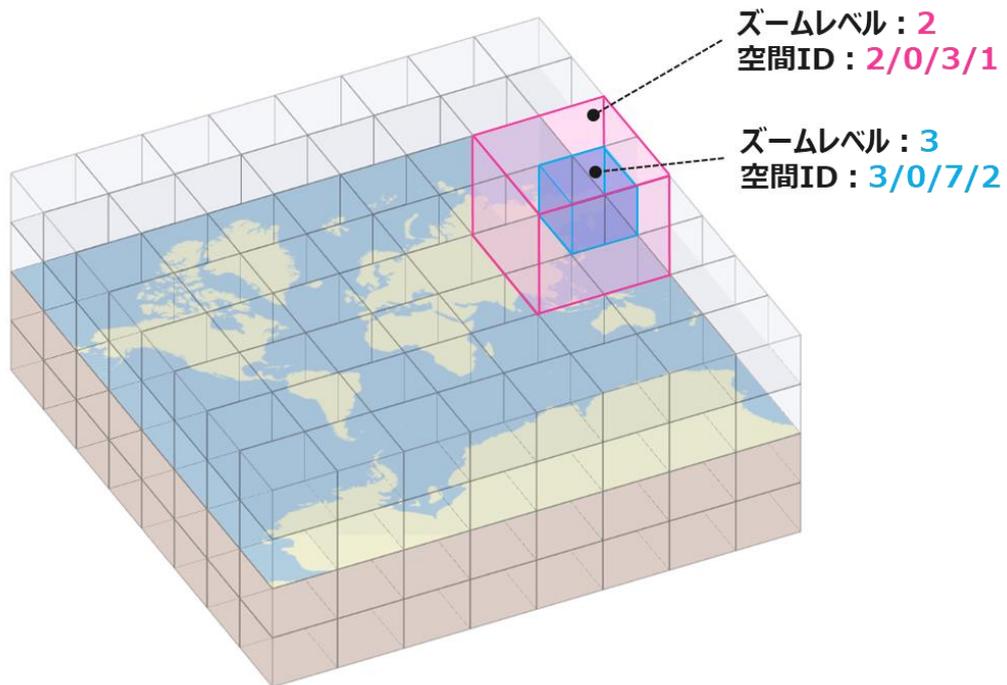


図 2-3 空間ボクセルのイメージ

2.3.2. 空間ボクセルによる空間分割方式の基本要素

空間ボクセルによる空間の分割方式を定義するための基本要素として、以下の 3 つの要素があり、次節以降で各要素の詳細について叙述する。

- (1) 空間ボクセルの高さの基準
- (2) 水平方向の空間分割方式
- (3) 鉛直方向の空間分割方式

空間分割方式の採用にあたっては、一般的に流通している地理空間情報及び関連するサービスとの親和性、ボクセルの形状・サイズの利便性、グローバルでの利用可能性等の点を考慮している。

(1) 空間ボクセルの高さの基準 (2) 水平方向の空間分割方式 (3) 鉛直方向の空間分割方式

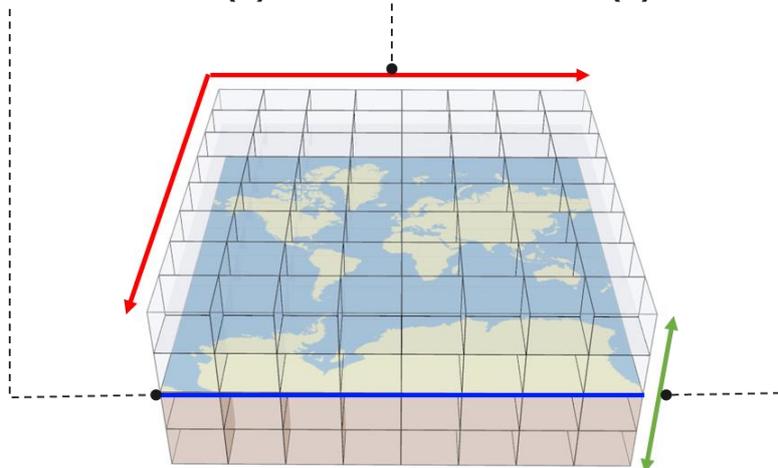


図 2-4 空間分割方式の基本要素

(1) 空間ボクセルの高さの基準

① 空間ボクセルを配置する基準面

地球上の3次元空間に空間ボクセルを配置する基準となる面はジオイド（地球の重力による位置エネルギーの等しい面）とし、標高 0m の面を平らな面とした座標空間上に空間ボクセルを配置する。

空間ボクセルの高さの値は標高値に相当し、標高を基準に整備された空間属性情報をそのまま空間ボクセルに紐付けできる。

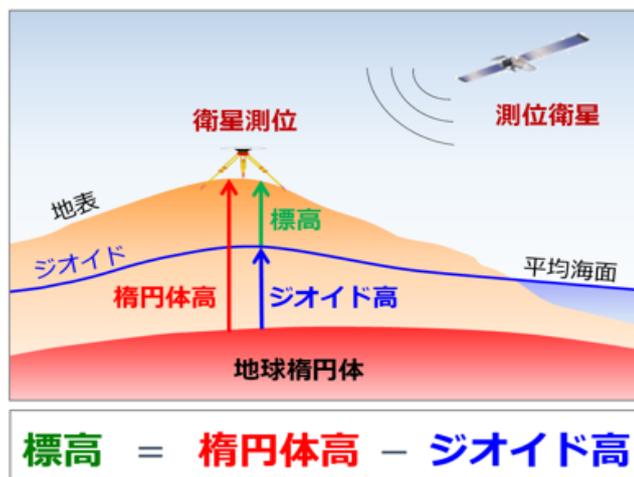


図 2-5 楕円体高、ジオイド高、標高の関係²（一部離島を除く³）

² 国土地理院 https://www.gsi.go.jp/buturisokuchi/grageo_geoid.html

³ 一部の離島においては基準面補正量を導入 ※2025年4月以降適用
<https://www.gsi.go.jp/sokuchikijun/hyoko2024rev.html#shikumi2>

② ジオイドモデル

衛星測位から高さを求めるためのデータとして「ジオイドモデル」がある。ジオイドモデルは国や地域によって使い分けられているため、高さの基準をメタデータとして明示するようにする。日本においては、東京湾平均海面に一致する等ポテンシャル面がジオイドと定められており、標準的な高さの基準面である。

(2) 水平方向の空間分割方式

① 水平方向の分割

水平方向の分割については、Webメルカトル図法⁴で定義されている地図範囲を全体範囲として行う。

対象とする地球上の領域は以下の範囲である。

経度 (X) : 西経 180 度～東経 180 度

緯度 (Y) : 南緯 85.0511 度～北緯 85.0511 度

上記範囲の領域をズームレベル 0 とし、ズームレベルが 1 つ増えるごとに 4 分割（南北方向に 2 分割、東西方向に 2 分割）を繰り返す。この分割方式は、多くの Web 地図サービスで採用されている XYZ タイル (Slippy map tilenames⁵) と同様である。

② 測地系

水平方向の分割のベースとなる地図の測地系は、世界測地系（日本測地系 2024 (JGD2024) または WGS84)⁶とする。必要に応じて、適用範囲に採用した測地系を空間 ID に紐付けたデータのメタデータとして明示できるようにする。

③ x インデックス、y インデックスの割り当て

分割された各図郭の識別番号として、経度（東西方向）の「x インデックス」と緯度（南北方向）の「y インデックス」を割り当てる。

北西端の端点が西経 180 度、北緯 85.0511 度である図郭における x と y のインデックスを (0, 0) として、x インデックスは東方向、y インデックスは南方向にかけて 1 つずつ値を増分する。

x インデックスの値

西端の図郭 : $x = 0$ 、東端の図郭 : $x = 2^{\wedge} \text{ズームレベル} - 1$

(西端の図郭の 0 から東方向の図郭にかけて 1 つずつ値を増分する)

⁴ Webメルカトル図法については、その地図範囲の定義を採用しているものであり、空間 ID 対応システムで利用する地図の投影法は任意である。空間 ID の算出には、世界測地系（日本測地系 2024 (JGD2024) または WGS84) の準拠楕円体上での緯度経度 (10 進法表記) を使用する。

⁵ Slippy map tilenames - OpenStreetMap Wiki

https://wiki.openstreetmap.org/wiki/Slippy_map_tilenames

⁶ 日本測地系 2024 (JGD2024) は 2025 年 4 月以降適用

y インデックスの値

北端の図郭 : $y = 0$ 、南端の図郭 : $y = 2^{\wedge} \text{ズームレベル} - 1$

(北端の図郭の 0 から南方向の図郭にかけて 1 つずつ値を増分する)

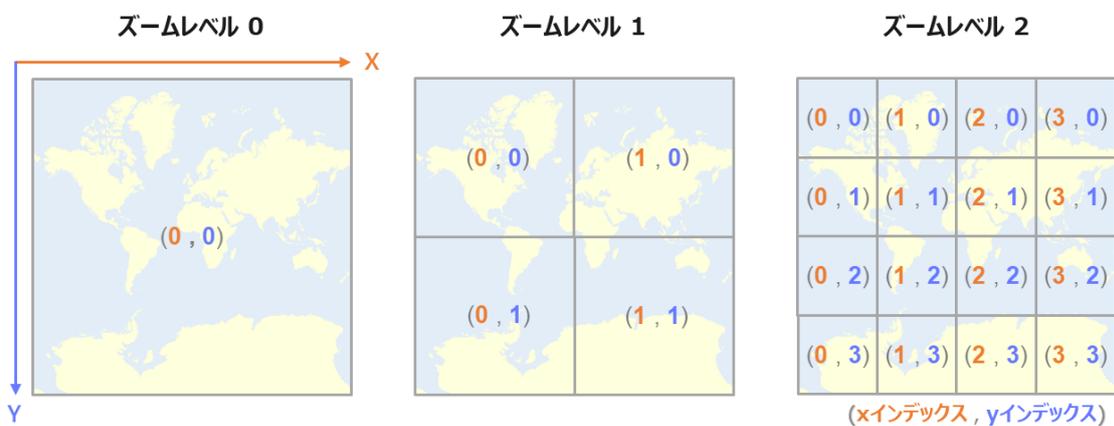


図 2-6 水平方向の分割とインデックスの割り当て

(3) 鉛直方向の空間分割方式

① 鉛直方向の分割

鉛直方向の分割については、高さのプラス方向及びマイナス方向の範囲をもとに行う。各範囲は以下のとおりである。

プラス方向の範囲 (標高値) : 0m ~ 33,554,432m

マイナス方向の範囲 (標高値) : - 33,554,432m ~ 0m

上記のプラス方向の範囲及びマイナス方向の範囲をズームレベル0とし、ズームレベルが1つ増えるごとに2等分割を繰り返す。

② f インデックスの割り当て

鉛直方向に分割された各範囲の識別番号として、「f インデックス」が割り当てられる。ズームレベルが1つ増えるごとに、プラス方向は、0 からプラス方向に1つずつ増分した値を割り当てる。マイナス方向は、-1 からマイナス方向に、1つずつ減分した値を割り当てる。

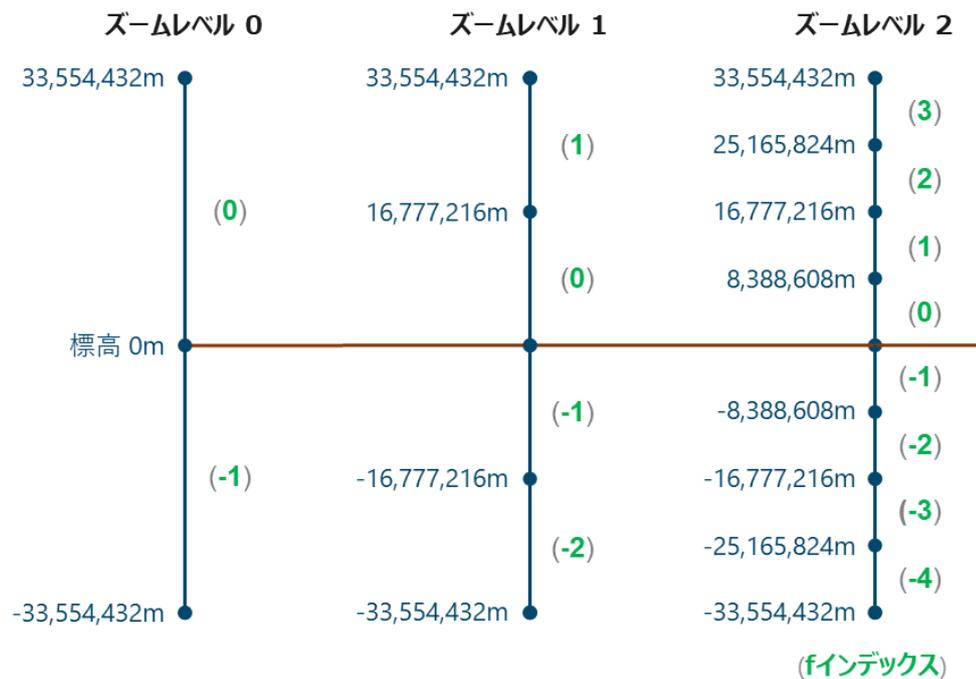


図 2-7 鉛直方向の分割とインデックスの割り当て

※ 分割点となる標高値は、上位側のインデックスに含まれる。

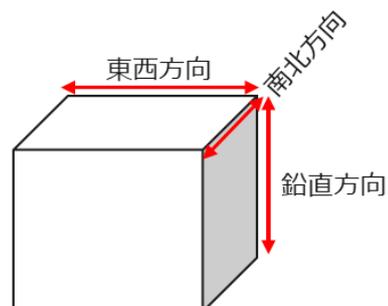
例) ズームレベル 2 の 8,338,608m の f インデックスは「1」

2.3.2.1. 空間ボクセルのズームレベルとサイズ

空間ボクセルは、ズームレベル 0 からズームレベルが 1 つ増えるごとに 8 分割を繰り返す、サイズが小さくなっていく。ズームレベル 0~26 における空間ボクセルの水平方向（東西方向・南北方向）、鉛直方向のサイズ例は以下のとおりである。なお、ズームレベル 27 以上も定義することはできる。

表 2-1 ズームレベル 0～26 における空間ボクセルのサイズ例

ズームレベル	水平方向		鉛直方向 (m)
	東西方向 (m)	南北方向 (m)	
0	40,075,016.68	40,075,016.68	33,554,432.00
1	20,037,508.34	20,037,508.34	16,777,216.00
2	10,018,754.17	10,018,754.17	8,388,608.00
3	5,009,377.09	5,009,377.09	4,194,304.00
4	2,504,688.54	2,504,688.54	2,097,152.00
5	1,252,344.27	1,252,344.27	1,048,576.00
6	626,172.14	626,172.14	524,288.00
7	313,086.07	313,086.07	262,144.00
8	156,543.03	156,543.03	131,072.00
9	78,271.52	78,271.52	65,536.00
10	39,135.76	39,135.76	32,768.00
11	19,567.88	19,567.88	16,384.00
12	9,783.94	9,783.94	8,192.00
13	4,891.97	4,891.97	4,096.00
14	2,445.98	2,445.98	2,048.00
15	1,222.99	1,222.99	1,024.00
16	611.50	611.50	512.00
17	305.75	305.75	256.00
18	152.87	152.87	128.00
19	76.44	76.44	64.00
20	38.22	38.22	32.00
21	19.11	19.11	16.00
22	9.55	9.55	8.00
23	4.78	4.78	4.00
24	2.39	2.39	2.00
25	1.19	1.19	1.00
26	0.60	0.60	0.50



上表の東西方向・南北方向の距離は緯度0度における距離を示しており、高緯度になるにしたがって距離が短くなる。

なお、鉛直方向の距離はいずれの経度・緯度においても一律で表 2-1 のとおりである。

参考例として、日本の各都市（那覇市、東京都、札幌市）のズームレベル 16～26 における空間ボクセルのサイズを以下に示す。

表 2-2 那覇市、東京都、札幌市のズームレベル 16～26 における空間ボクセルのサイズ例

那覇市 (那覇市役所付近) 緯度：北緯 約26.21度				東京 (東京都庁付近) 緯度：北緯 約35.89度				札幌市 (札幌市役所付近) 緯度：北緯 約43.06度			
ズーム レベル	水平方向		鉛直方向 (m)	ズーム レベル	水平方向		鉛直方向 (m)	ズーム レベル	水平方向		鉛直方向 (m)
	東西方向 (m)	南北方向 (m)			東西方向 (m)	南北方向 (m)			東西方向 (m)	南北方向 (m)	
16	548.98	546.01	512.00	16	497.22	495.01	512.00	16	447.48	445.86	512.00
17	274.49	273.00	256.00	17	248.61	247.51	256.00	17	223.73	222.92	256.00
18	137.24	136.50	128.00	18	124.31	123.75	128.00	18	111.86	111.46	128.00
19	68.62	68.25	64.00	19	62.15	61.88	64.00	19	55.93	55.73	64.00
20	34.31	34.13	32.00	20	31.08	30.94	32.00	20	27.97	27.87	32.00
21	17.16	17.06	16.00	21	15.54	15.11	16.00	21	13.98	13.93	16.00
22	8.58	8.53	8.00	22	7.77	7.73	8.00	22	6.99	6.97	8.00
23	4.29	4.27	4.00	23	3.88	3.87	4.00	23	3.50	3.48	4.00
24	2.14	2.13	2.00	24	1.94	1.93	2.00	24	1.75	1.74	2.00
25	1.07	1.07	1.00	25	0.97	0.97	1.00	25	0.87	0.87	1.00
26	0.54	0.53	0.50	26	0.49	0.48	0.50	26	0.44	0.44	0.50

※ 水平方向の長さはGRS80楕円体上の測地線距離。
空間ボクセルの水平面上の上辺(北側)と下辺(南側)で
東西方向の距離が異なる場合は、下辺の距離を記載。



© Earthstar Geographics

図 2-8 那覇市、東京、札幌市の位置

2.4. 空間 ID

2.4.1. 空間 ID の形式

空間 ID の構成要素には、z, f, x, y があり、各要素をスラッシュ (/) で連結した配列とする。

空間 ID の構成要素

- {z} : ズームレベル
- {f} : 標高 (鉛直方向) インデックス
- {x} : 経度 (東西方向) インデックス
- {y} : 緯度 (南北方向) インデックス

空間 ID の配列

{z}/{f}/{x}/{y}

例 : 20/1/931369/413142

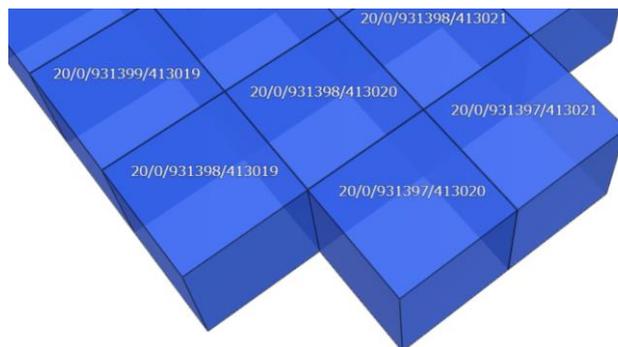


図 2-9 各空間ボクセルの空間 ID を表記したイメージ

2.4.2. 空間 ID の各インデックスの算出方法

経度、緯度、標高、ズームレベルから空間 ID の各インデックス (f, x, y) を算出するための計算式は、以下のとおりである⁸。

lng: 経度 [10 進度]
lat: 緯度 [10 進度], *lat_rad*: 緯度 [ラジアン]
h: 標高 [m]
z: ズームレベル

$n = 2^z$
 $Z = 25$ (ボクセルの高さが 1m となるズームレベル)
 $H = 2^Z$ [m]
 $f = \text{floor}\left(n * \frac{h}{H}\right)$

⁸ 国連ベクトルタイルツールキット <https://github.com/unvt/zfxy-spec>

$$x = \text{floor} \left(n * \left(\frac{\text{lng} + 180}{360} \right) \right)$$

$$y = \text{floor} \left(\frac{n}{2} * \left(1 - \frac{\log \left(\tan(\text{lat_rad}) + \left(\frac{1}{\cos(\text{lat_rad})} \right) \right)}{\pi} \right) \right)$$

2.4.3. 高さ情報を持たないデータに対応した空間 ID の形式

空間属性情報には、高さ情報を持たない 2 次元データ（例：人口統計、用途地域、土地利用）が数多くある。これらのデータを空間 ID に紐付ける場合、標高を示す f インデックスは不要であることから、f インデックスを除いた以下の ID 形式を適用することとする。

高さ情報を持たないデータに対応した空間 ID の配列

{z}/{x}/{y}

例：20/931369/413142

これは地理院地図などの Web 地図サービスの XYZ タイル番号と同じ配列である。

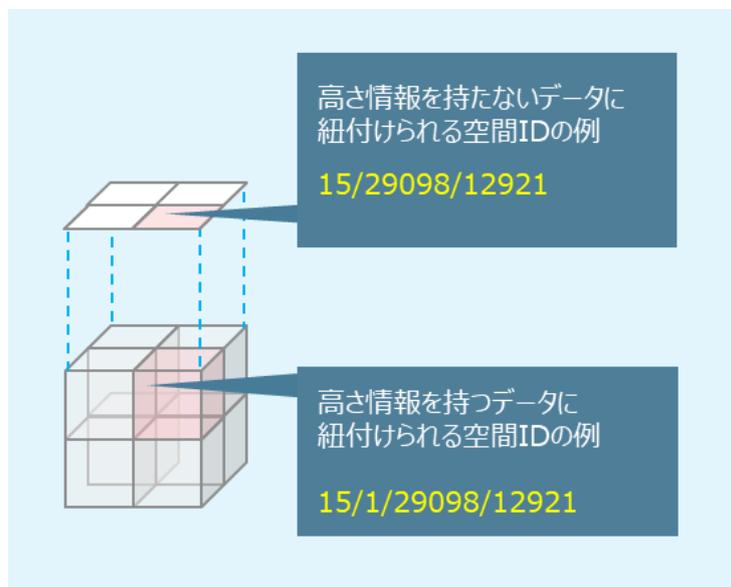


図 2-10 高さ情報を持たないデータに対応した空間 ID の例

2.5. 拡張仕様

本節では、空間 ID の役割を局所的な空間情報の管理にも適用するための「ローカル空間 ID」及び、時間に応じて変化/移動する情報にも適用するための「時空間 ID」の仕様について示す。なお、ローカル空間 ID および時空間 ID は拡張オプションのため、適用は任意であり、ユースケースのニーズに応じて実装要否を判断する。

2.5.1. ローカル空間 ID

ローカル空間 ID は、局所的な範囲を対象とした情報共有が必要となる場合の空間情報を管理するための共通ルールとして利用できるものである。具体的には、建物などの屋内においてローカル座標系で管理されているデータを扱う場合や、エンタメユースケースにおいて現実空間とデジタル空間を連携して様々な地点で空間を活用する場合、また、車両の荷台の内部などのように対象となるローカル座標系自体が移動する場合等にローカル空間 ID を適用することを想定している。

本項では、ローカル空間 ID との明示的な対比のために、通常の「空間 ID」を「グローバル空間 ID」と表記する。

(1) ローカル空間 ID の空間分割方式

ローカル空間 ID の水平方向に対する空間分割方式は以下のとおりである。

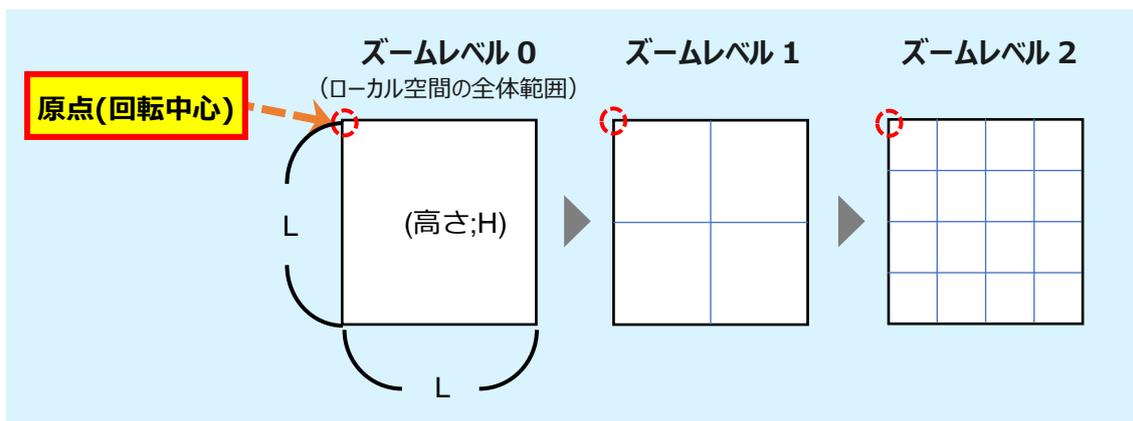


図 2-11 ローカル空間 ID の空間分割方式(水平方向)

グローバル空間 ID と同じ概念・ロジックに基づく相互運用性、コードの再利用性などからローカル空間の全体範囲を定義し、グローバル空間 ID と同じ方式で、2次元空間では4分割、3次元空間では8分割を繰り返す。

ローカル空間 ID の鉛直方向に対する空間分割方式は以下のとおりである。

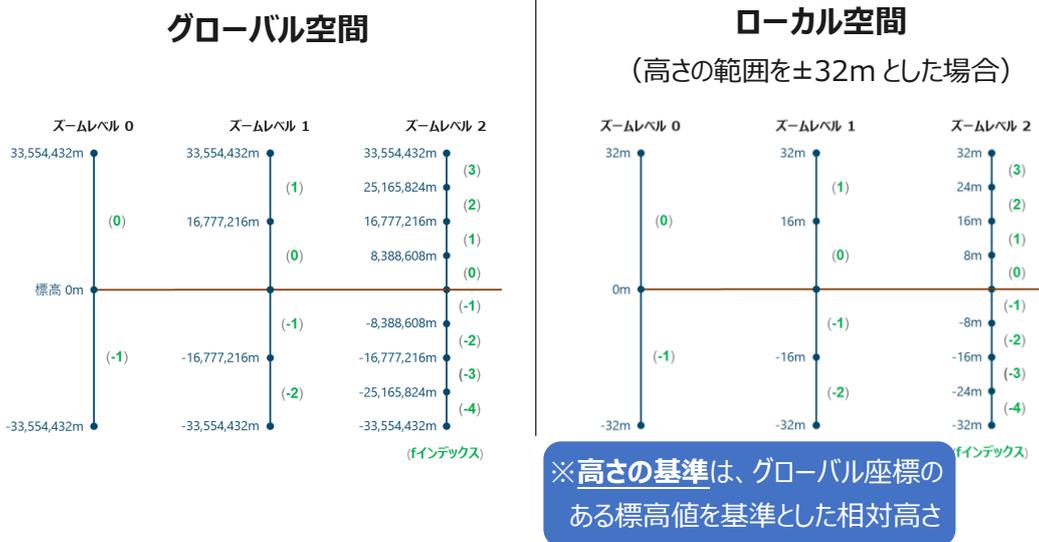


図 2-12 ローカル空間 ID の空間分割方式(鉛直方向)

(2) ローカル空間 ID の形式

ローカル空間 ID の構成要素には、z, f, x, y があり、各要素をスラッシュ (/) で連結した配列とする。形式としてはグローバル空間 ID と同じ形式となる。

空間 ID の構成要素

- {z} : ズームレベル
- {f} : 高さインデックス
- {x} : X 軸インデックス
- {y} : Y 軸インデックス

空間 ID の配列

{z}/{f}/{x}/{y}
例: 2/0/1/3

各空間ボクセルに対するローカル空間 ID を表記したイメージは以下のとおりである⁹。この例では「ズームレベル 2」を示しており、2次元では「4 × 4セル」の配列になっている。

⁹ ローカル空間 ID SDK デモ <https://geolonia.github.io/local-spatial-id-js-sdk/>



図 2-13 各空間ボクセルのローカル空間 ID を表記したイメージ

(3) ローカル空間 ID の各インデックスの算出方法

ローカル空間の座標値(X, Y)と高さ、ズームレベルからローカル空間 ID の各インデックス (f, x, y) を算出するための計算式は、以下のとおりである。

【ローカル空間 ID の各インデックスを算出する計算式】

$$x = \text{floor}(n * X / L)$$

$$y = \text{floor}(n * Y / L)$$

$$f = \text{floor}(n * h / H)$$

※ $n = 2^z$ (z: ズームレベル値)

(X, Y) = ローカル空間の座標値[m]

h = (グローバル座標系における原点の標高値を基準とした)高さ[m]

[L, H]: 全体範囲(L; 水平, H; 鉛直)の一辺の長さ

f: 鉛直方向インデックス

x: X 方向インデックス

y: Y 方向インデックス

ローカル空間 ID における座標軸と全体範囲の定義は以下のとおりである。

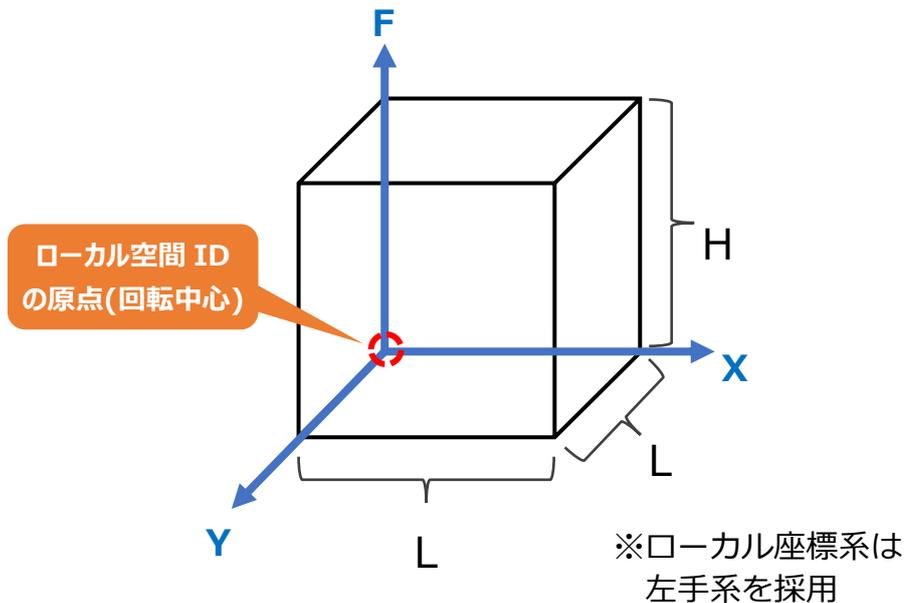


図 2-14 ローカル空間 ID における座標軸と全体範囲の定義

ローカル座標系は左手系を採用する。図中では、ローカル空間 ID の原点を赤丸で示し、この点がグローバル座標に対して定位する際の回転中心となる。また、ローカル空間 ID の全体範囲の一辺の長さは、水平方向は L(X, Y は共通)、鉛直方向は H として独立に定義できるものとする。

ローカル空間 ID の計算式に基づく試算例は以下のとおりである。



図 2-15 ローカル空間 ID の計算式に基づく試算例

ローカル空間 ID の仕様では水平(X, Y)のインデックスの起点はローカル座標系の原点を基準としており、ローカル空間 ID 仕様の計算式に基づき、シンプルな設定値を用いて試算した結果を例示する。ローカル空間 ID の計算式に基づく試算例(全体範囲を[32×32×32m]の立方体ボクセルとした場合)は以下のとおりである。

[点 A]…全体範囲の範囲内のローカル座標系の最大座標の点&高さ値が0付近の1ボクセル目

⇒ (X, Y, h)=(31.5, 31.5, 0.5)

[点 B]…全体範囲の正の範囲外の1ボクセル外 ⇒ (X, Y, h)=(32.5, 32.5, 32.5)

[点 C]…全体範囲の負の範囲外の1ボクセル外 ⇒ (X, Y, h)=(-0.5, -0.5, -0.5)

(4) ローカル空間 ID の座標定義

ローカル空間 ID の座標定義は、以下の要素で構成される。座標定義情報を規定することでグローバル空間 ID への対応付けが可能となる。

- ① 全体範囲の各辺の長さ[m 単位]
⇒水平方向(X, Y)は共通の長さ[L]とする。
鉛直方向は[L]とは異なる長さ[H]を定義できる。
- ② 原点の位置*(緯度・経度・標高)
⇒全体範囲の左上隅を座標原点及び回転中心にする。
- ③ 回転角*(オイラー角 ; $-180 \sim +180^\circ$) [度単位]
⇒[Z 軸まわり (X-Y 平面) の回転角 (ϕ)]
*ローカルのみ利用の場合、定義は任意とする。

※ローカル座標自体が移動するケースでは、移動中に動的に座標を更新せず、移動が完了した時点で②と③の情報を更新するという運用も考えられる。

ローカル空間 ID の全体範囲を座標定義情報に基づいて、グローバル空間 ID と重ね合わせた例は以下のとおりである。

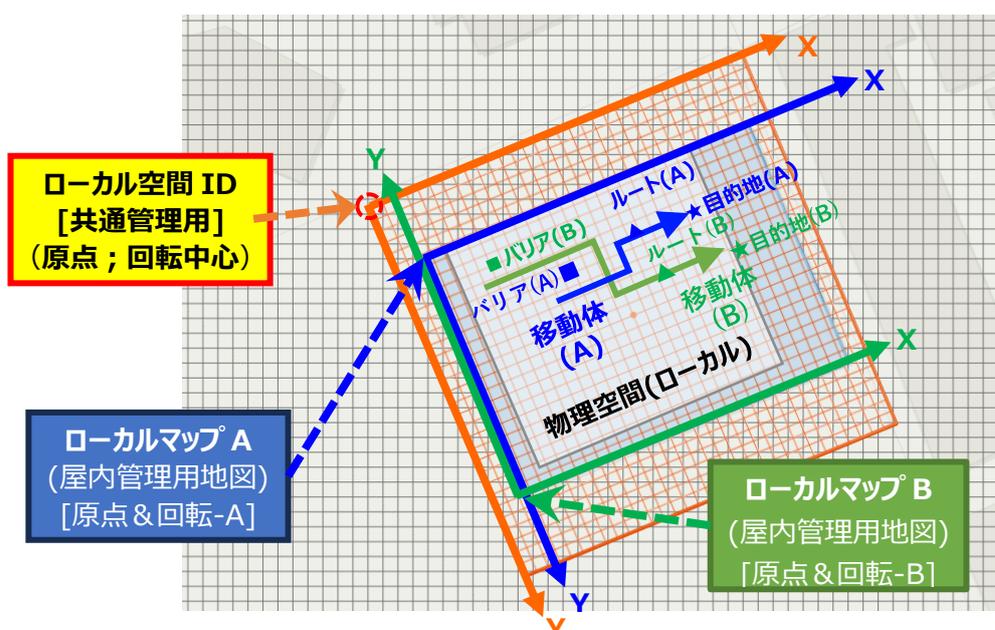


図 2-16 ローカル空間 ID の座標定義情報に基づくグローバル空間 ID と重ね合わせ

(5) ローカル空間 ID のズームレベルとボクセルサイズ

ローカル空間 ID のズームレベルとボクセルサイズの関係は以下のとおりである。

ズームレベル	全体範囲	ボクセルサイズ[m]	ズームレベル	全体範囲	ボクセルサイズ[m]
0	32	32	0	25.6	25.6
1	32	16	1	25.6	12.8
2	32	8	2	25.6	6.4
3	32	4	3	25.6	3.2
4	32	2	4	25.6	1.6
5	32	1	5	25.6	0.8
6	32	0.5	6	25.6	0.4
7	32	0.25	7	25.6	0.2
8	32	0.125	8	25.6	0.1

図 2-17 ローカル空間 ID のズームレベルとボクセルサイズ

上記の左側の例では、1 辺が 32m の全体範囲として定義した場合、ズームレベル[5]が 1m ボクセルとなり、運用上の基準ユニットとすることができる。基準ユニットを『0.1m(10cm)』としたい場合には、全体範囲の 1 辺の長さを 25.6m とすることでズームレベル[8]が 10cm ボクセルになる。つまり、運用上の基準ユニットを任意に規定したい場合には、全体範囲のサイズを『基準ユニットの 2 のべき乗』と規定することで対応できる。

(6) ローカル空間 ID の範囲外の取扱い

ローカル空間 ID の全体範囲の範囲外の取り扱いについては以下のとおりである。

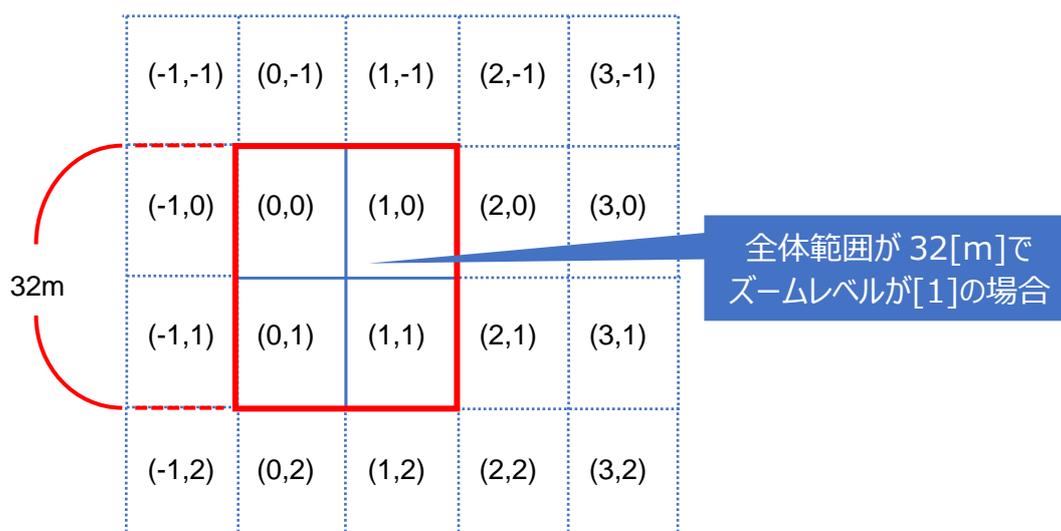


図 2-18 ローカル空間 ID の範囲外の取扱い

定義した全体範囲の外側の取扱いは、定義した外側(マイナスにも)にもインデックスは付与できる。ただし、ライブラリなどのツール類の影響を考慮して、全体範囲の外側の ID は利用不可とする。

全体範囲の左右境界の繋ぎ処理については、グローバル空間 ID の左右方向(定義域; 西経 180° ~ 東経 180°)の境界でボクセルの隣接関係を判定する場合、両端を隣接しているとして繋いでいる。ただし、ローカル空間 ID では左右の両端を繋がない(上下方向と同様の扱いとする)。

(7) ローカル空間 ID とグローバル空間 ID の対応付け

3D ビューアによるローカル空間 ID とグローバル空間 ID の対応付けの動作例は以下のとおりである。

(1) ローカル空間 ID/共通ライブラリ

ここでは、デモサイトを利用したローカル空間 ID の座標定義情報に基づく 3D ビューアによる表示例を以下に示す。

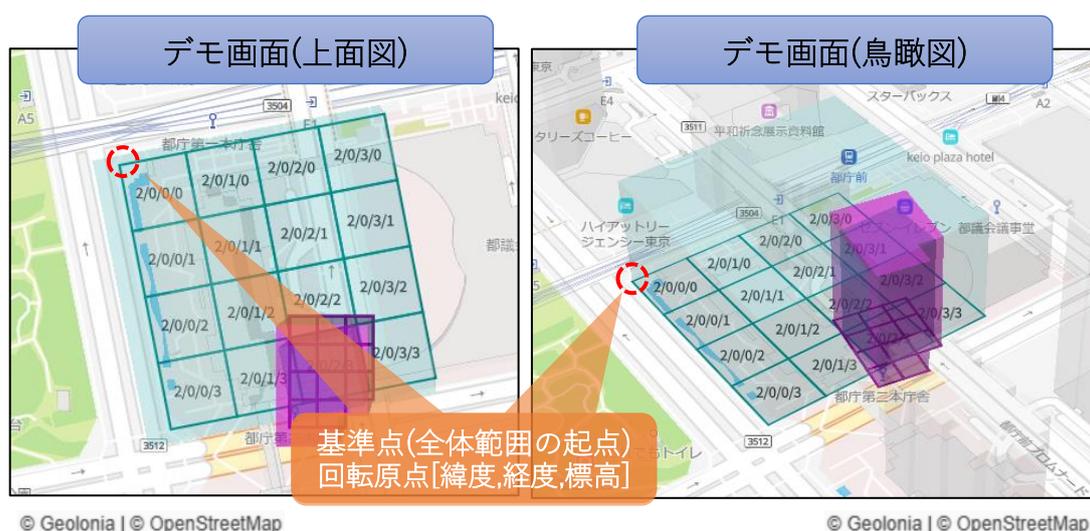


図 2-19 ローカル空間 ID の 3D ビューアによる表示例

また、同じくデモサイト上では以下の設定及び操作を行うことができる。ローカル空間 ID の一部を選択することで、対応するグローバル空間 ID のボクセル(図中の紫色のボクセル)を取得することができる。



図 2-20 デモサイトの設定画面及び情報確認画面

上記の設定画面では、座標定義情報管理サイトにて取得した座標定義情報を用いて、以下の情報を設定し、関連する情報を画面上で確認することができる。

- 基準点の[緯度, 経度][標高][角度]
- 全体範囲のサイズ(水平/高さ)
- モードの選択 (ローカル or グローバル)
- ズームレベル[ローカル, グローバル]

2.5.2. 時空間 ID

空間 ID が持つ空間情報を検索・統合する検索キーとしての役割を、時間に応じて変化/移動する情報にも適用できるように時空間 ID を定義する。時間軸を一定間隔で区切った時間間隔ごとの時空間 ID に、移動体や事象の情報を紐付けて統合することにより、特定の時空間における概況（移動体数、人流、気象状況など）の把握が可能となる。これにより混雑度やリスク度の高い時空間の特定などに役立てることができる。

(1) 時空間 ID の構成

時空間 ID は空間 ID と時間 ID から構成され、時間 ID はオプションで追加できるものとする。

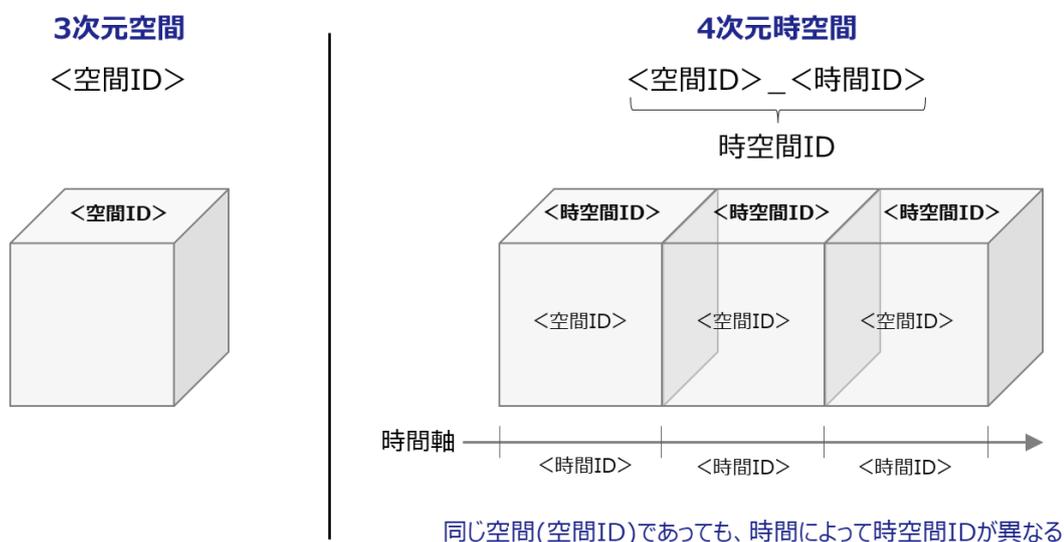


図 2-21 空間 ID と時空間 ID の比較イメージ

(2) 時間軸の分割方式

時間軸の起点 [1970 年 1 月 1 日 0:00] から一定の時間間隔 [任意指定 (単位: 秒)] で時間軸を分割し、その時間間隔ごとに一意の識別子を付与することとする。時間軸の分割における定義項目とその内容は以下のとおりである。

時間軸の起点: 1970/1/1 0:00 (協定世界時(UTC))

時間の単位: 秒

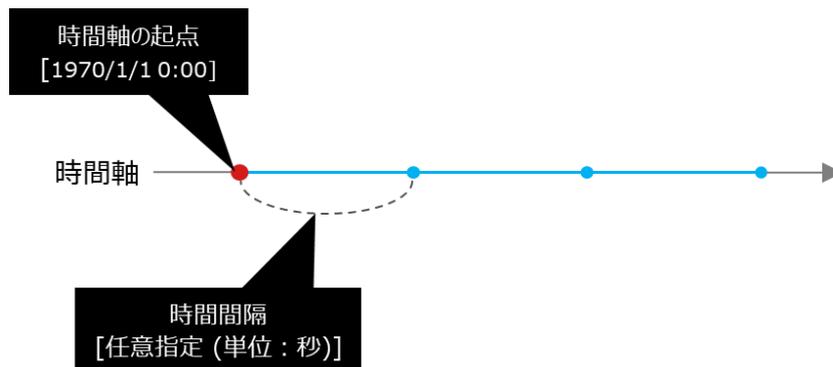


図 2-22 時間軸の分割

時間軸の起点は UNIX 時間の起点である 1970/1/1 0:00 (協定世界時(UTC)) を採用しており、起点からの経過秒数は UNIX 時間に相当する。UNIX 時間は国や地域による時差やサマータイムの影響を受けないため、グローバルで統一した時間 ID を定義できる。UNIX 時間は協定世界時 (UTC) であることから、時間 ID の算出に使用する時間は協定世界時 (UTC) に統一する。

時間間隔は任意の秒数を指定できることとする。複数レベルを定義することが可能であり、空間 ID のズームレベルとは独立している。

(3) 時間インデックスの算出方法

UNIX 時間と時間間隔 (秒) から各時間間隔の識別子である時間インデックスを算出するための計算式を以下に示す。

t : 時間インデックス (各時間間隔の識別子)
 u : UNIX 時間 (1970 年 1 月 1 日 0:00 からの経過秒)
 i : 時間間隔 (秒)

$$t = \text{floor}(u / i)$$

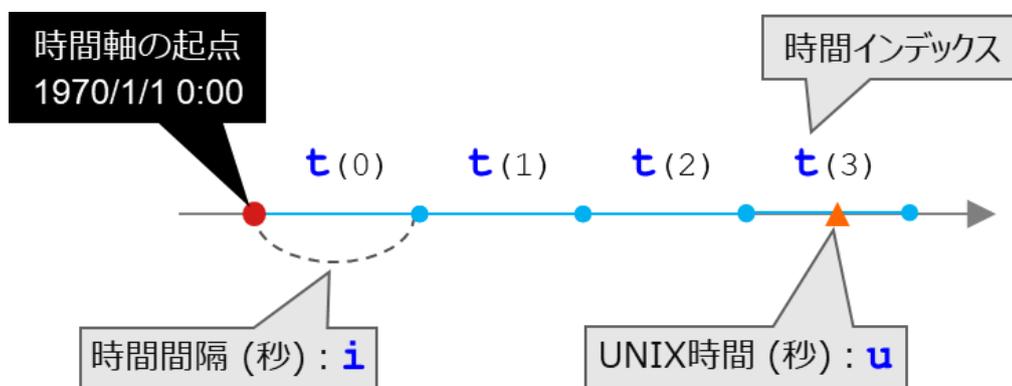


図 2-23 分割された時間軸における各要素

(4) 時空間 ID の形式

空間 ID ($\{z\}/\{f\}/\{x\}/\{y\}$) に時間 ID (時間間隔と時間インデックスから構成) を付加したものを時空間 ID とする。

空間 ID と時間 ID を連結する文字は、アンダースコア ($_$) とする (空間 ID では各インデックスの連結文字としてスラッシュ ($/$) を使用しているが、高さ情報を持たないデータに対応した空間 ID ($\{z\}/\{x\}/\{y\}$) を利用した際との混同を避けるため、代表的な連結文字のひとつであるアンダースコアを採用)。

時間 ID の構成要素

{i} : 時間間隔 (秒)

{t} : 時間インデックス

時空間 ID の配列

$\{z\}/\{f\}/\{x\}/\{y\}_\{i\}/\{t\}$

例 : 12/0/3638/1614_1800/809712

(5) 異なる時間間隔定義の時間 ID 間の関係性把握

時間 ID の要素から計算をして秒単位の時間範囲を求めることで、異なる時間間隔定義の時間 ID 間の関係性 (重複・包含関係など) を把握することが可能となる

時間 ID の時間範囲の始点 (秒) の計算式

$$i * t \quad [\text{時間間隔} * \text{時間インデックス}]$$

時間 ID の時間範囲の終点 (秒) の計算式

$$(i * t) + t \quad [(\text{時間間隔} * \text{時間インデックス}) + \text{時間間隔}]$$

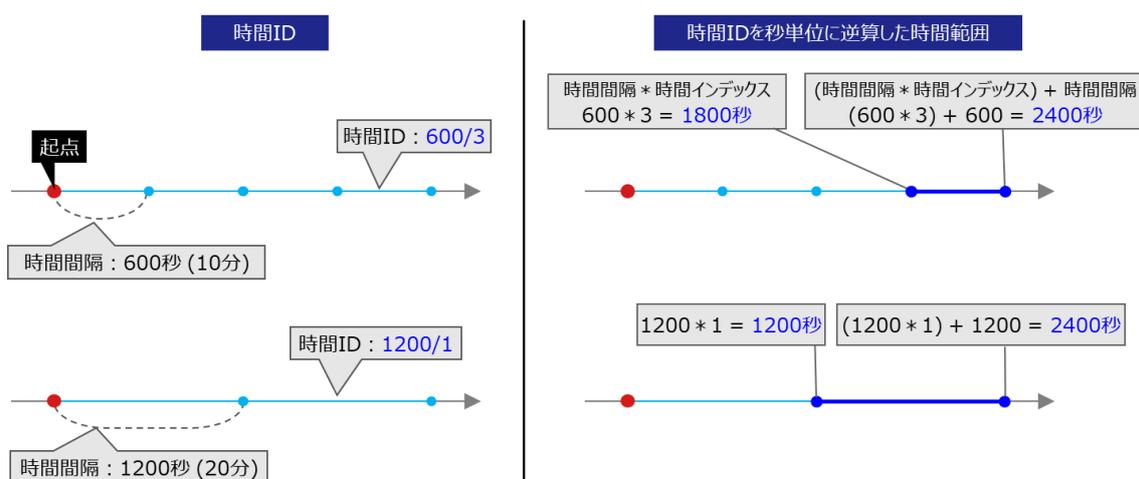


図 2-24 時間 ID から秒単位の時間範囲への変換

3. 空間 ID の使用方法

空間 ID は前章に記載のとおり仕様を規定しているが、実際に活用するにはユースケースに応じて様々な実装パターンがある。本章では、空間 ID の基本的な使い方について記述するとともに、実証等の成果を踏まえて検討してきた空間 ID を活用したシステムの実装例を記載する。

3.1. 空間 ID の基本的な使い方

3.1.1. 識別子としての空間 ID

空間 ID は、地物等のデータを検索する際の識別子としての役割を持つことができる。例えば、リクエストされた特定の空間 ID について、該当する空間 ID の空間ボクセルに含まれる地物等のデータを抽出して配信する、といった利用が想定される。

空間 ID 対応システムが各種サービスを提供する際、データ配信事業者が有する様々な種類の情報の集約において、空間 ID を識別子として使用することで、統一形式による情報収集の効率化や、新規データ配信事業者の参入促進が期待される。(図 3-1)

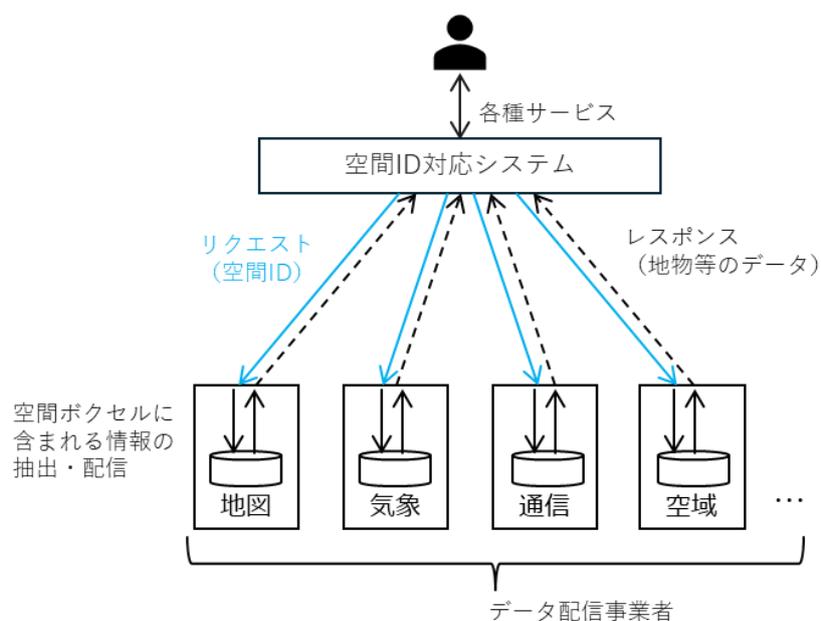


図 3-1 空間 ID の識別子としての利用

3.1.2. 空間ボクセルへの属性付与

各空間ボクセル内に存在する地物等の情報を空間 ID と紐付けて、空間 ID をインデックスとして用いることで、各情報を配信する際の空間的な単位を空間ボクセルとすることにより、統一された基準による情報流通が可能となり、各情報の検索や統合が容易になる。また、各情報を空間 ID 単位で抽象化することで空間の概況を把握することもできる。

各空間ボクセルと地物等の関係は、1 対 1、1 対多、多対 1 の場合がある。これらをデータスキーマとして定めることは行わないが、各関係における情報の付与・関連付けの基本的な考え方を以下に示す。

(1) 空間ボクセルと地物の関係が1対1の場合

1つの空間ボクセルに1つの地物等が含まれる場合は、空間ボクセルに対して地物等の属性情報を関連付けする。また、地物等の属性に空間IDが付与される。

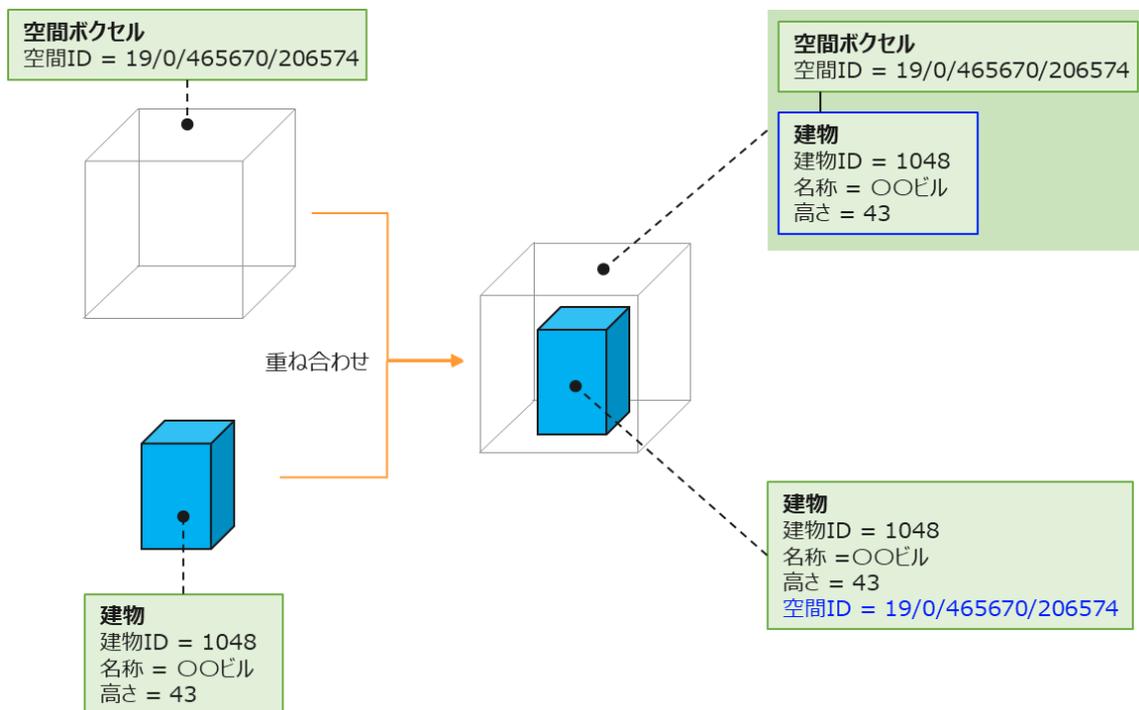


図 3-2 空間ボクセルと地物データの紐付け例 (1対1)

(2) 空間ボクセルと地物の関係が1対多の場合

1つの空間ボクセルに複数の地物等が含まれる場合は、空間ボクセルの各地物等の属性情報を付与又は関連付けする。地物等の属性に数値情報が含まれる場合は、その統計情報（個数、最大値、最小値、平均値等）を付加することも想定される。また、各地物等の属性に空間IDが付与される。

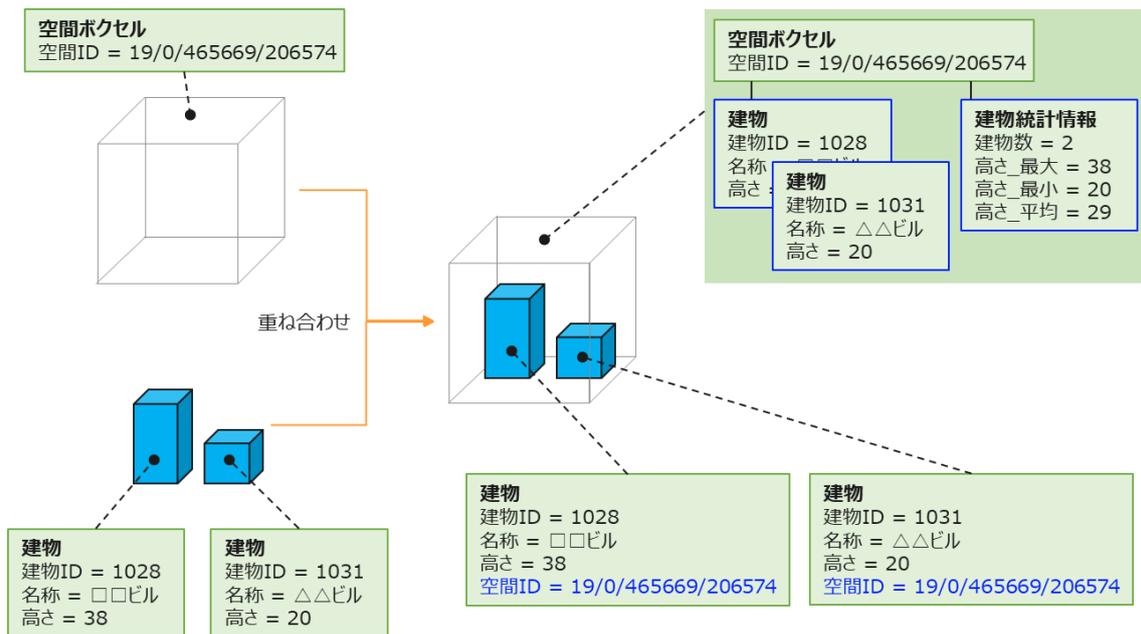


図 3-3 空間ボクセルと地物データの紐付け例 (1 対多)

(3) 空間ボクセルと地物の関係が多対 1 の場合

1 つの地物が複数の空間ボクセルに跨がる場合は、複数の空間ボクセルに対して同じ地物の属性情報を付与又は関連付けする。また、地物等を含む全ての空間ボクセルの空間 ID が地物等の属性に付与される。

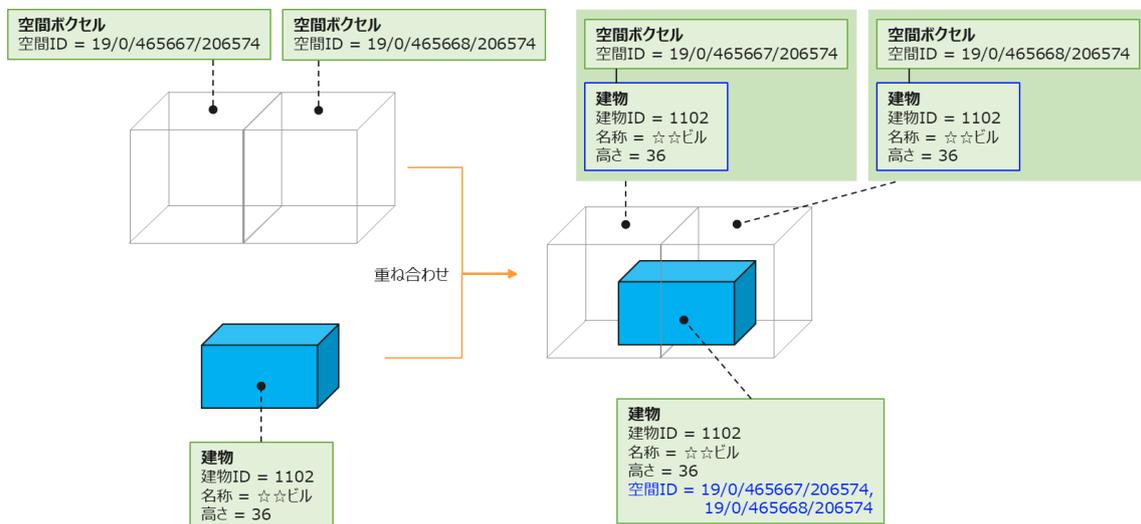


図 3-4 空間ボクセルと地物データの紐付け例 (多対 1)

3.1.3. 時間情報の扱い

地物等のデータの中には、属性として時間情報 (年/月/日/時刻) を含む場合がある。時間情報には瞬間的な情報と期間的な情報が存在する。瞬間的な情報は、地物等のある時点の状況 (予測も含む) が時刻とともに示される (例: 気象情報)。期間的な情報は、地

物等の実在期間や有効期間が開始時刻と終了時刻によって示される（例：道路工事情報）。時間情報を3次元空間のIDと関連付けることによって、空間と時間を用いた管理や検索が可能となる。



図 3-5 時間情報の利用例

時空間 ID も地物等の時間情報を利用するための仕組みであるが、一定の時間間隔ごとに情報を紐付けて統合することが主目的である。任意の時刻や期間の情報をもとにした管理や検索を行う場合は、属性として時間情報を利用することが適していると言え、ユースケースに応じた方式を適用する。

3.1.4. 空間ボクセルに地物等の情報を紐付ける際のズームレベルの選択

地物等のデータを付与又は関連付けする空間ボクセルのズームレベルの選択は任意であり、利用目的や求められる精度などに応じてズームレベルを選択する。また、1種類の地物データセット（例：建物データ）に対して複数のズームレベルの空間ボクセルを混在させて紐付けることも可能である。

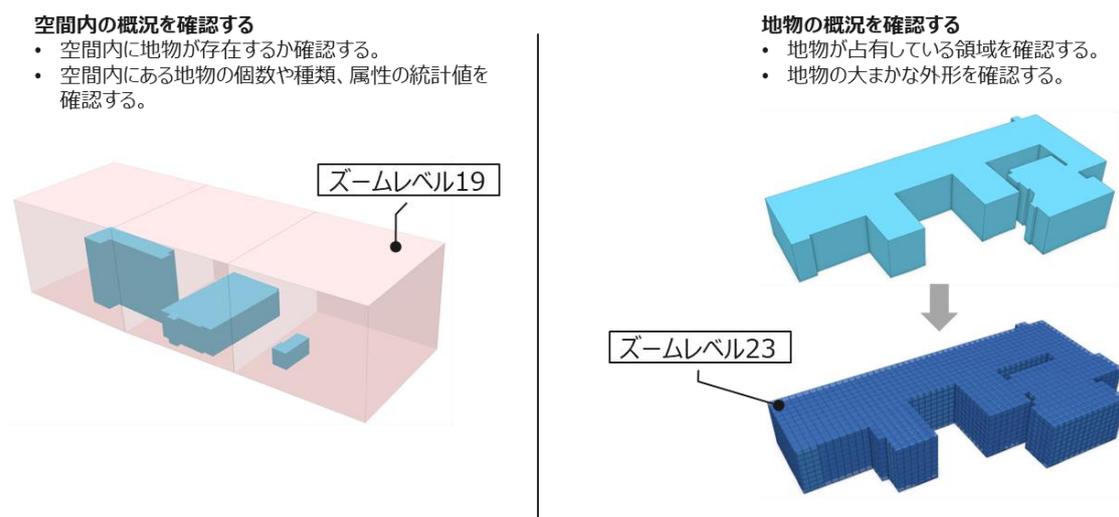


図 3-6 ズームレベルの設定例

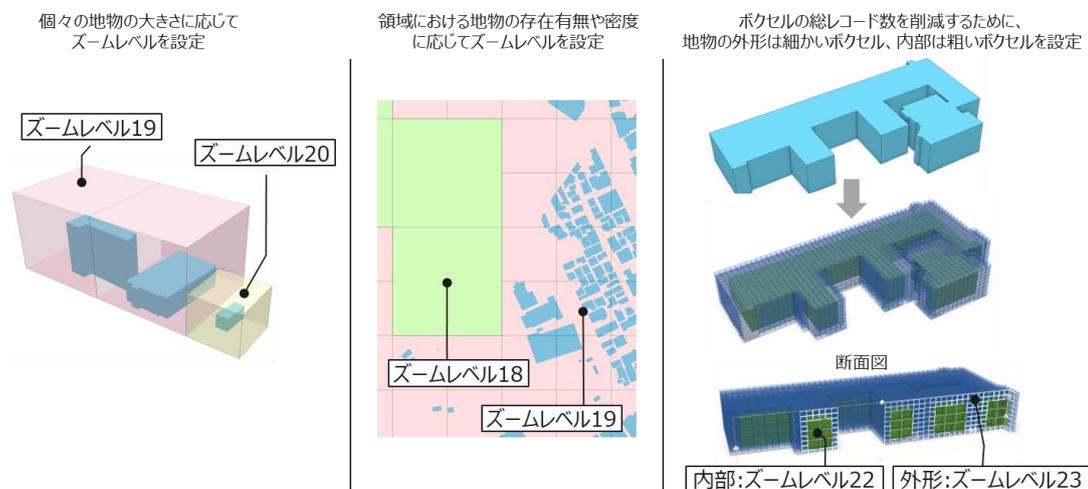


図 3-7 複数のズームレベルを混在させて設定する例

3.2. 空間 ID 対応システム

前節に空間 ID の基本的な使い方を記載したが、業務で使用するには、空間 ID を活用した機能をシステムとして実装する必要がある。

本節では、空間 ID 対応システムを構成する構成要素と、代表的な構成例を示す。なお、本節で示すシステム及び機能はあくまで使用例であり、実装を限定するものではないことに留意されたい。

実際のシステムを構築する際には、各事業者がシステム全体のアーキテクチャの中で空間 ID 対応システムの各構成要素の機能を適切に配置するものとする。また、これまでの実証等の成果を元に OSS を公開しており、システム構築の参考とすることができる。OSS の情報については次節に記載する。

3.2.1. システム概要

空間 ID 対応システムとは、2 章で定義した空間ボクセル及び空間 ID を利用して、異なる基準に基づいた多様な空間属性情報（地形情報、空域情報、気象情報、人流情報等）を様々なユースケースで利用するためのシステムの総称である。

なお、他分野とのデータ連携に係るプロトコルや共通機能の設計については、ウラノス・エコシステム・データスペースズ リファレンスアーキテクチャモデル (ODS-RAM)¹⁰ を参照すること。今後、ODS-RAM に基づきデータスペースが整備されることで、利用者は、空間範囲（地理空間）及びデータ項目等の条件を指定して、該当する空間属性情報を取得して統合する作業をデジタル完結・自動化することが可能となる。図 3-8 に空間 ID 対応システム及び本ガイドラインの対象範囲を示す。

¹⁰ ウラノス・エコシステム・データスペースズ リファレンスアーキテクチャモデル ホワイトペーパー <https://www.ipa.go.jp/digital/architecture/reports/ouranos-ecosystem-dataspaces-ram-white-paper.html>

空間 ID 対応システムが集約・提供する情報は、利用者の用途（ユースケース）別に大きく異なる。さらに、ユースケースごとに要求される性能やデータに求められる信頼性も異なることから、特定のエリアで特定の空間属性情報を提供する「空間情報サービス」が複数分散配置され、全体として空間範囲及び空間属性情報の網羅性を確保するオープンな分散構造とする。

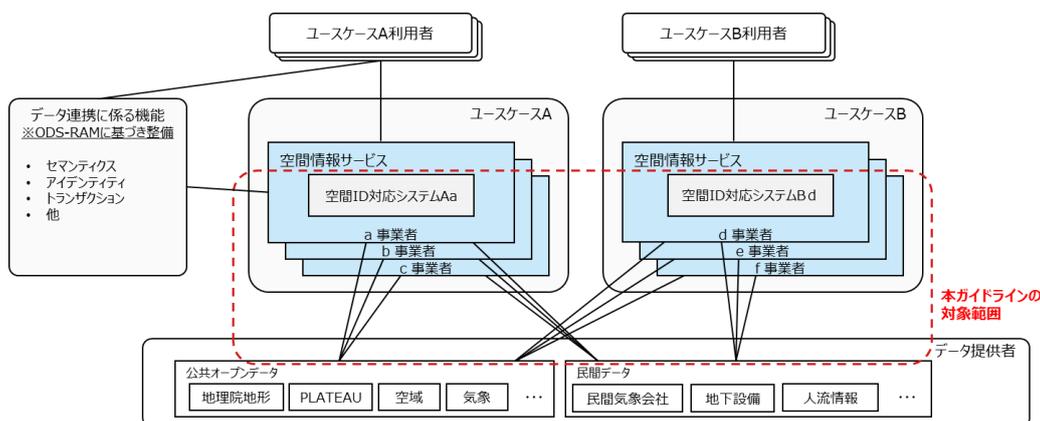


図 3-8 空間 ID 対応システム及び本ガイドラインの位置付け

(1) 空間情報サービス

空間 ID 対応システムを運用して、利用者からの空間範囲とデータ項目に関するリクエストに応じて、該当する空間属性情報（該当する空間ボクセル及びその属性値の集合）をリターンするサービスを空間情報サービスという。

(2) 空間 ID 対応システム

空間 ID に係る機能を実装し、利用者からのリクエストに応じたリターンを行うシステム。リターンを行うために必要となる公共オープンデータや民間データ等の情報を収集、集約する。空間 ID 対応システムの基本構造は 3.2.2 節に記載する。

(3) セマンティクス

分散して存在する空間情報や他分野のサービスについて、サービス提供事業者に関する基本情報、アクセス方法、サービス提供範囲、期間及びデータ項目等の情報を登録・管理し、利用者が必要とする情報を取得するためにアクセスすべきサービスを検索する仕組みをカタログシステムという。ODS-RAM においてセマンティクスレイヤとして構成され、宛先の問題及び意味の問題を解決するレイヤである。利用者は、セマンティクスを用いることで、多様な情報を提供するサービスが分散して存在している状況であっても、必要な情報を取得する手段を知ることが可能となる。

(4) アイデンティティ

空間情報や他分野のサービスにおいて取り扱うデータのなりすまし防止、改ざん防止、属性情報の証明を担保するとともに、アクセス権限を管理するルール及び仕組みの総体を認証基盤という。ODS-RAM においてアイデンティティレイヤとして構成され、認証の問題

及び認可の問題を解決するレイヤである。空間情報サービスで提供するデータが、実空間において安全性・信頼性を確保するために使われるユースケースにおいても利用される。

(5) トランザクション

空間情報や他分野のサービス間でデータをやり取りし、サービス提供を行う際のリクエスト、レスポンス、及びデータ形式に係る相互運用を行うための仕組み。ODS-RAM においてトランザクションレイヤとして構成され、形式の問題及び要求の問題及び手段の問題を解決するレイヤであり、宛先・意味解決がなされた結果に基づき、データ転送のプロセス制御が行われる。

3.2.2. 空間 ID 対応システムの基本構造

空間 ID 対応システムは、空間情報サービスが利用者の必要とする空間属性情報の空間範囲及びデータ項目に応じて、空間属性情報を提供する仕組みである。

空間 ID 対応システムは、企業・個人・学術団体・公的機関を問わず、空間属性情報と空間 ID を紐付けていれば、その他は自由に構築しサービスを提供できる分散型でオープンなシステムであり、設計・実装は個々の空間情報サービス提供者に委ねる。他方で、空間 ID 対応システムごとにアクセス方法が異なれば、利用者の利便性が損なわれる。そのため、3.2.3 節に示すとおり API 設計方針を定め、各空間 ID 対応システムの連携の効率化を促進するものとする。空間 ID 対応システムの機能配置例を図 3-9 に示す。各構成要素の概要は以下の通りである。

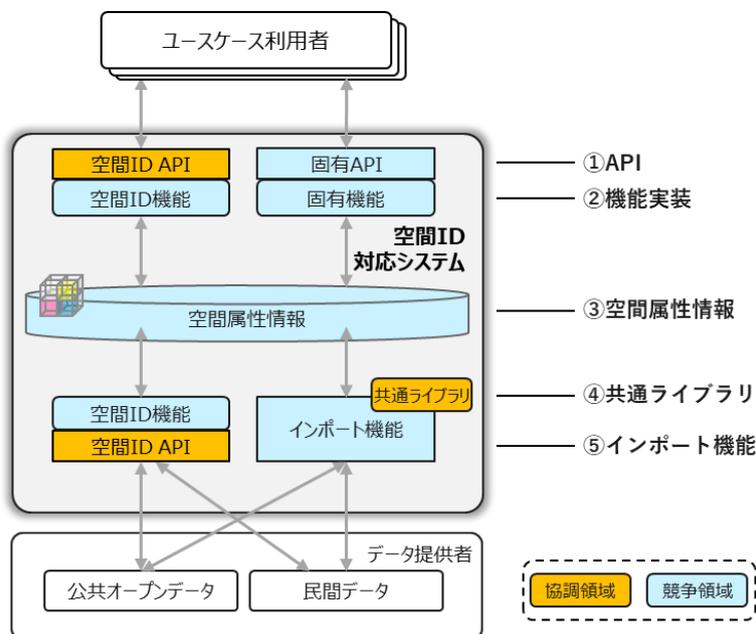


図 3-9 空間 ID 対応システムの機能配置例

(1) API

利用者が共通的に使用する機能については、利用者のシステムと空間 ID 対応システムを繋ぐインターフェースの設計方針を 3.2.3 節に記載する。インターフェースは、

空間 ID 対応システム間やデータ提供者のシステムを繋ぐ際にも使用される。

(2) 機能実装

API 設計方針に基づく機能および固有機能について、各事業者が機能をプログラムとして実装する。

(3) 空間属性情報

空間 ID 対応システムは、空間 ID 対応システムが利用者に提供する空間属性情報の空間範囲（地域）及びデータ項目（地形、気象等）に関する情報を保有する。本例では空間属性情報を空間 ID 対応システム内で保有する例を示す。一方で、空間 ID 対応システム内では空間属性情報を保有せずに、データ提供者のシステムに①に記載の API を実装して、利用者のシステムとデータ提供者のシステムを直接的に繋ぐという例も想定される。

(4) 共通ライブラリ

3.2.4 節で示すとおり、空間 ID 対応システムを設計・実装するうえで、使用する頻度が高いと想定される機能を共通ライブラリとして整備し、OSS として提供する。

(5) インポート機能

既存のファイルデータに空間 ID を紐付けることで空間属性情報を生成するインポート機能の考え方及び実装例を 3.2.5 節に示す。

3.2.3. API 設計方針

(1) API の概要

空間 ID 対応システムにおいて、利用者が「同一機能は同一の手順で空間属性情報を生成・参照・更新・削除」することを目的として、利用頻度が高いと考えられる機能について、API の設計方針を記す。実装にあたっては、用途に応じてパラメータを拡張した API を定義することとし、操作については本ルールに従うことを推奨する。本ルールに記載しない用途の操作については、各実装で拡張して定義する。API の概略を図 3-10 に示す。

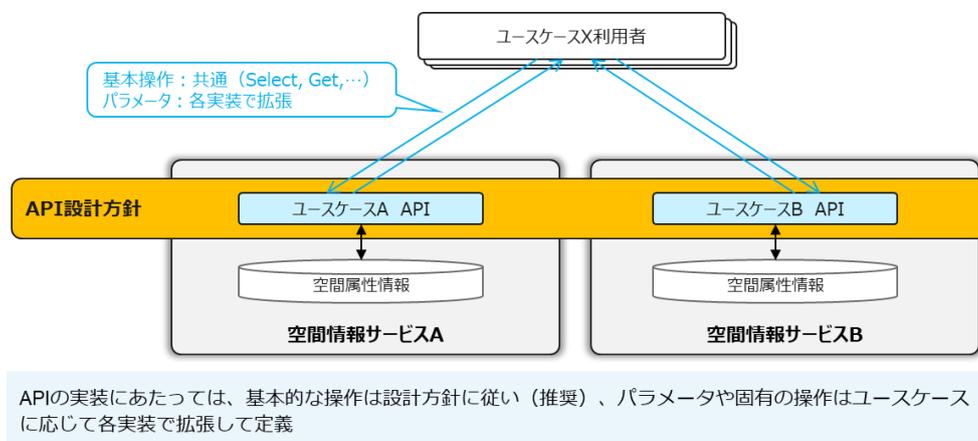


図 3-10 API の概略

(2) API の命名方針

API の名称は、特定の機能を一意に識別できるように定義する必要がある。すなわち、同一の機能であるにも関わらず複数の API の名称を持つことや、異なる機能であるにもかかわらず同一の API の名称を持つことを回避する必要がある。そのため、個別のユースケース毎の固有識別子、機能識別子等を組み合わせて特定の機能を一意に識別できるように、以下のとおり API の命名方針を記す。

なお、API の名称が重複することを回避するための仕組みや、API 拡張に伴う運用については、各ユースケースで規定する。

① 用語

- データ識別子
提供する情報のまとまりを定義する識別子。
単一形とし、複数形にしない。（表記形式は適用するプロトコルに依存する）

- メジャーバージョン
API のメジャー番号。v{メジャーバージョン番号}の形式とする。

- 機能識別子
検索、取得、登録・更新及び削除の機能を示す識別子。
その他の機能を設計する場合は、任意に名前を規定する。
 - ・ Select 空間ボクセル検索を伴う値の取得
 - ・ Get 空間ボクセル検索を伴わない値の取得
 - ・ Put 値の登録・更新
 - ・ Create 値の登録
 - ・ Update 値の更新
 - ・ Delete 値又は空間ボクセル全体の削除

- 操作対象識別子
API ごとに任意に規定する、機能の操作対象を表す識別子。
主に操作対象とする空間の形状や対象を名前とすることを推奨する。
定義例： Server, ReserveArea

② プロトコル共通

- プロダクト名
API のプロダクトマーケティング名として、API、UI、ドキュメント、利用規約、請求明細、商用契約などで一貫して使用する名前を定義する。
形式： {固有識別子} {データ識別子} API
定義例： UAS Airspace API

- サービスホスト名
1つ以上のネットワークアドレスに解決可能な DNS 名（RFC1035 に基づく）等の文字列を定義する。

すべて小文字とするが、それ以外の形式は規定しない。

定義例： example.co.jp

③ gRPC

- パッケージ名
プログラム名と一貫した名前を定義する。
すべて小文字とする。
形式： {固有識別子}. {データ識別子}. {メジャーバージョン}
定義例： uas.airspace.v1
- インターフェース名
『service {データ識別子}API』と定義する。
- メソッド名
rpc 名がメソッドである。
アッパーキャメルケースとする。
形式： {機能識別子} {操作対象識別子}
定義例： GetReserveArea、PutReserveArea、DeleteReserveArea
- 入力情報名
リクエストメッセージの名前を定義する。以下のいずれかで定義する。
 - ・ {メソッド名}Request
 - ・ 空のメッセージ (google.protobuf.Empty を使用)
 - ・ リソースタイプ
 - ・ オペレーションを表すリソース
- 出力情報名
レスポンスメッセージの名前を定義する。以下のいずれかで定義する。
 - ・ {メソッド名}Response
 - ・ 空のメッセージ (google.protobuf.Empty を使用)
 - ・ リソースタイプ
 - ・ オペレーションを表すリソース
- その他メッセージ名
アッパーキャメルケースで定義する。
短く簡略な名称を定義することを推奨する。

④ REST

- URI
リソースを一意に識別子するための URI をパターンごとに定義する。
共通的なルールは以下の通りである。
 - ・ URI のキーワードはケバブケース (ハイフンで連結) で表記し、ア

ンダースコアは使用しない。

- ・ URI はすべて英数字を用いて表記する。
- ・ URI 内の英字はすべて小文字で表記する。

1. 実行パターン (Command パターン)

コマンドを実行するパターンの際に定義する。HTTP メソッドは POST。

URI 形式： {サービスホスト名}/{サブパス}/{固有識別子}/api/{略称 API 名}/{メジャーバージョン}/{コマンド名}

{サブパス}

サービスホスト名に続くサブパス。存在する場合のみ定義する。

{略称 API 名}

API 名の略称。{データ識別子}の小文字の形式とする。

{コマンド名}

実行するコマンドの名称。動詞または動詞＋名詞（例：connect-server）の形式とする。

URI 定義例：

https:// example. co. jp/uas/api/airspace/v1/connect

2. 作成パターン (Create パターン)

リソースを作成するパターンの際に定義する。HTTP メソッドは POST。

URI 形式： {サービスホスト名}/ {サブパス}/ {固有識別子}/api/{略称 API 名}/{メジャーバージョン}/{リソースパス}

{リソースパス}

リソース名とリソースを一意に特定するためのキー情報を「/」で連結したパス。リソースはデータベースのエンティティ等のアクセス情報資産のことであり、名詞の複数形で定義する。

URI 定義例：

https://example. co. jp/uas/api/airspace/v1/spatial-information

3. 取得パターン (Read パターン)

リソースを取得するパターンの際に定義する。HTTP メソッドは GET。

なお、リソースを取得する際に任意の項目で絞り込む検索条件がある場合はクエリパラメータを指定する。

URI 形式： {サービスホスト名}/ {サブパス}/ {固有識別子}/api/{略称 API 名}/{メジャーバージョン}/{リソースパス}?{クエリパラメータ}

※?以降はリソースを検索によって絞り込む場合のみ

{クエリパラメータ}

検索条件やページ番号などサーバに情報を送信するための文字列（変数）。パラメータ名はローワーキャメルケースで指定する。

URI 定義例：

<https://example.co.jp/uas/api/airspace/v1/spatial-information/1>

<https://example.co.jp/uas/api/airspace/v1/spatial-information/1?xxx=aaa&yyy=bbb>

4. 更新パターン (Update パターン)

リソースを更新するパターンの際に定義する。HTTP メソッドは PUT。

URI 形式： {サービスホスト名}/ {サブパス}/ {固有識別子}/api/{略称 API 名}/{メジャーバージョン}/{リソースパス}

URI 定義例：

<https://example.co.jp/uas/api/airspace/v1/spatial-information/1>

5. 削除パターン (Delete パターン)

リソースを削除するパターンの際に定義する。HTTP メソッドは DELETE。

URI 形式： {サービスホスト名}/ {サブパス}/ {固有識別子}/api/{略称 API 名}/{メジャーバージョン}/{リソースパス}

URI 定義例：

<https://example.co.jp/uas/api/airspace/v1/spatial-information/1>

- プロパティ
リクエストボディやレスポンスボディのパラメータ。JSON の場合、ローワーキャメルケースで定義する。

(3) API データモデル

空間情報の基本的なデータモデルは図 3-11 のとおり、モデルのクラス（建造物や空域など）によって、モデルオブジェクトやボクセルのプロパティ（ボクセルの値）は異なる。クラスによって、プロパティは異なるが、1 モデルオブジェクトが複数のボクセルを所有する構造は変わらない。

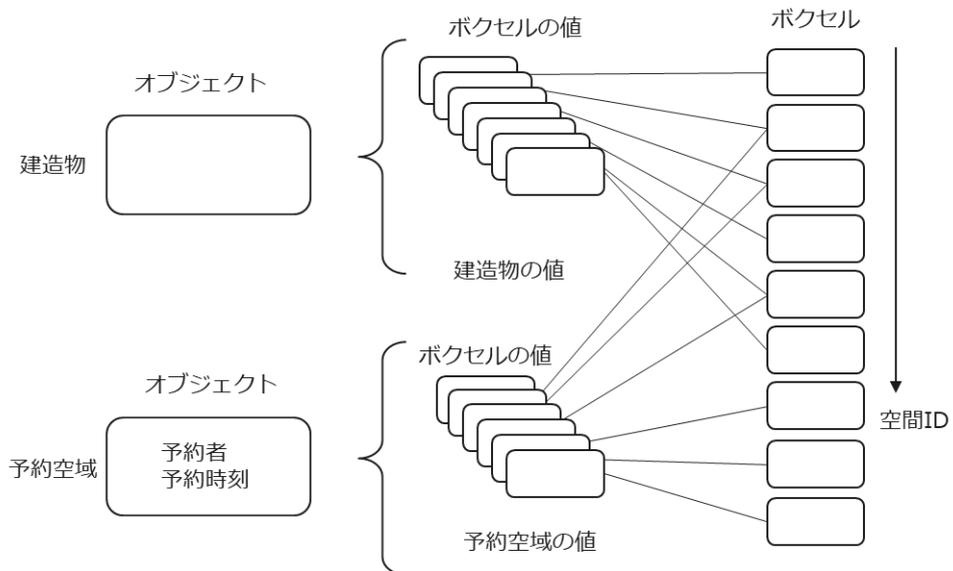


図 3-11 基本的なデータモデル

基本的なデータモデルに基づき、図 3-12 に示す API を規定する。

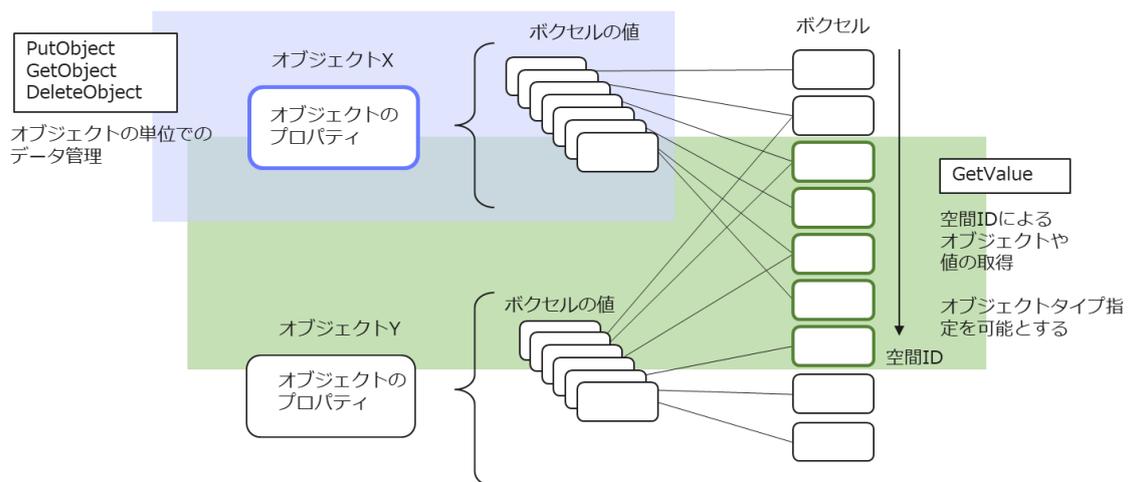


図 3-12 API の基本構成

(4) API データ仕様

空間情報のタイプ及び値を表 3-1 に示す。なお、表 3-1 はあくまで例であり、ユースケースに応じてタイプや値を適宜実装する。

表 3-1 空間 ID 対応システムにおける空間情報の例

空間情報データ識別子		属性値 (値)
バリア	地形情報	情報元リンク
	建物情報	情報元リンク
一般	気象	現況、予報、情報元リンク
	電波強度	キャリアごと電波強度、情報元リンク
	人流情報	情報元リンク

(5) API 仕様

空間 ID 対応システムが標準的に備えるべき API を表 3-2 に示す。

表 3-2 空間 ID 対応システムにおける標準 API

分類	API 名	機能	入力	出力
空間情報 管理	登録 (PutObject)	オブジェクトを登録する	空間情報	ObjectID
	取得 (GetObject)	オブジェクトの空間情報を取得する	ObjectID	空間情報
	削除 (DeleteObject)	オブジェクトを削除する	ObjectID	
	値取得 (空間 ID 指定) (GetValue)	指定領域の空間情報を取得する	空間 ID 群	空間情報

以下、各 API の機能概要を示す。また、API 仕様の例を APPENDIX-2 に示す。

① オブジェクトの登録・取得・削除を行う API

空間 ID 群で表現されるバリアや飛行空域などの空間情報について、登録・取得・削除を行う API である。バリアや飛行空域はオブジェクトとして空間 ID 群で表現される。オブジェクト登録によって、登録した空間 ID 群に対して、オブジェクト ID が付与される。

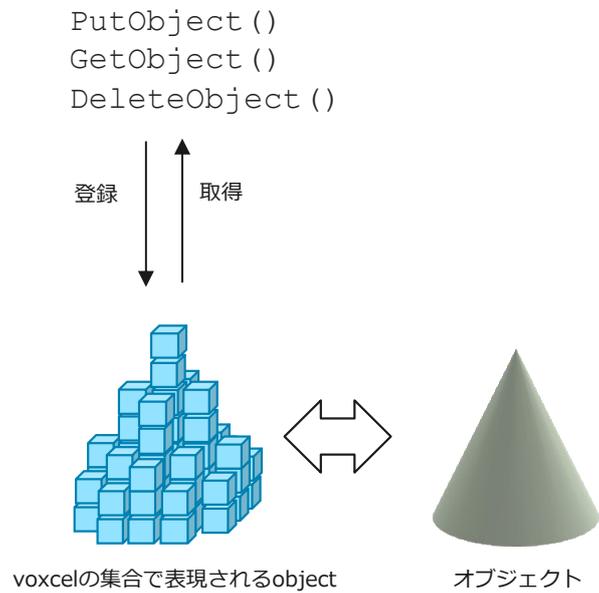


図 3-13 オブジェクト登録削除API

- ② 指定領域の空間情報を取得するAPI
 領域と空間情報タイプを指定し、該当する空間情報を得るためのAPIである。

例

オレンジ色枠のエリアおよびタイプに飛行制限を指定すると、飛行制限空域の情報が返る

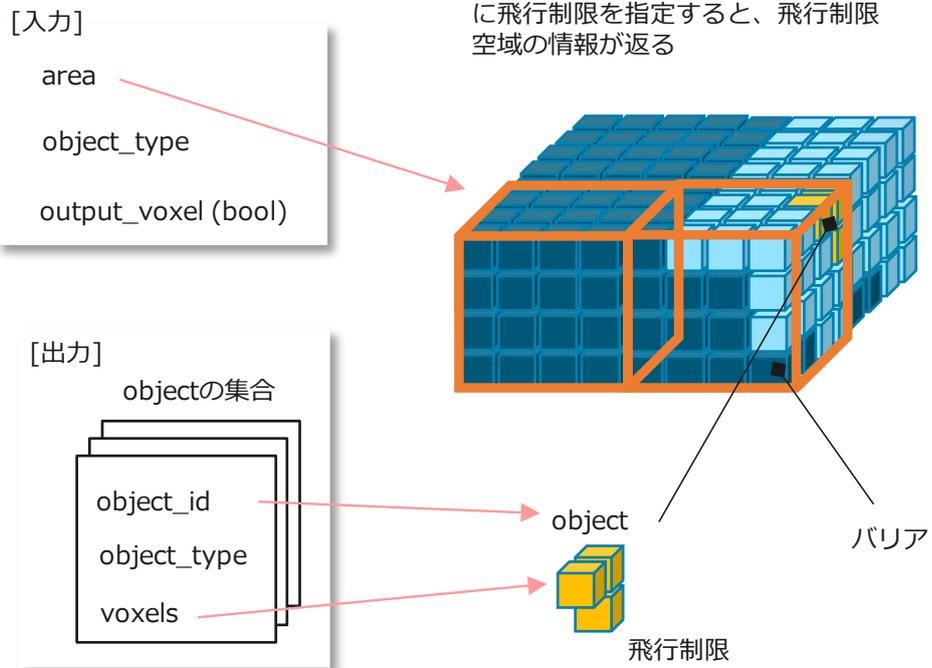


図 3-14 空間情報取得API

3.2.4. 空間 ID 変換に係る機能

(1) 基本的な考え方

空間 ID 対応システムと利用者システム・データ提供者間の相互運用性や開発生産性を高めるため、ユースケース横断で頻出する共通的な機能については、その機能を実現するプログラムの共通ライブラリを整備し、OSSとして公開する。想定される処理のうち、空間 ID 及び空間 ID の集合に関する演算を行う処理を共通ライブラリの対象としている。

<機能実装におけるライブラリとの関係>

<ライブラリ整備の考え方>

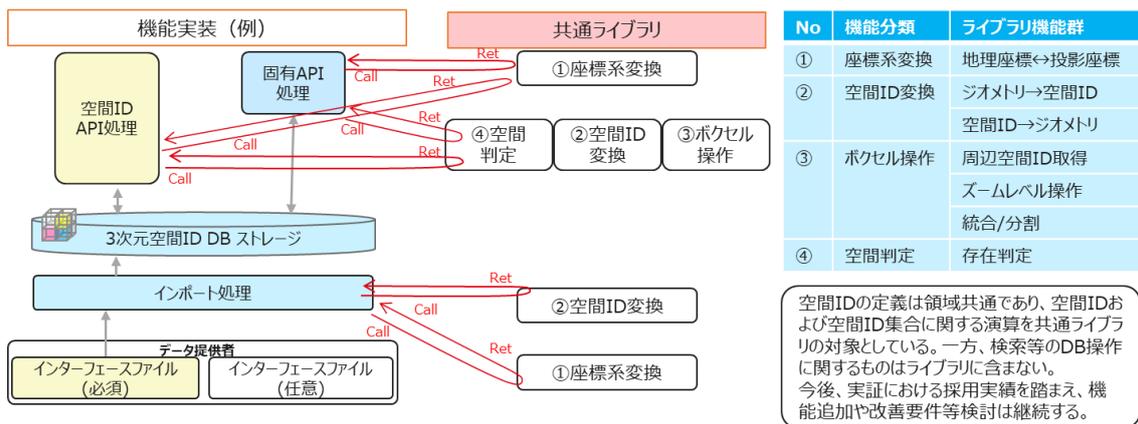


図 3-15 機能実装におけるライブラリとの関係

① 座標変換

データ提供者から空間 ID 対応システムに対して提供される座標を持つデータについて、空間 ID を扱う事前・事後処理として、地理座標系と投影座標系の変換を行う。

② 空間 ID 変換

空間 ID 対応システムの空間属性情報のデータベースに格納するために、緯度、経度、高さで表現されるジオメトリ情報を空間 ID に変換する。この際、点座標を空間 ID に変換するだけでなく、体積を持つ形状については空間 ID に対応するボクセル形状との衝突判定を行うことで、重なりを持つ空間 ID を全て出力する。また、反対に、空間 ID に対応するボクセルの形状情報をジオメトリ情報として出力する。なお、この場合、元の体積を持つ形状の情報は失われている。

③ 空間ボクセル操作

ジオメトリ情報を空間 ID に変換した後に、周辺の空間 ID の取得や、空間ボクセルのズームレベルの操作並びに空間 ID の統合及び分割といった空間ボクセルの操作を行う。

④ 空間判定

ジオメトリ情報が、指定した空間 ID に含まれるか否かを判定する。

(2) 共通ライブラリ機能概要

(1) で示した共通ライブラリの概要における各機能の概要を表 3-3 に示す。

なお、共通ライブラリは実証の成果を元に python 版および Node.js 版の 2 種類を提供するが、それぞれの実証で対象としたユースケースにより機能に差異があるため、機能概要を確認し、適宜、参考としてシステムに実装されたい。

表 3-3 共通ライブラリ機能概要

#	機能概要	共通ライブラリ Python版	共通ライブラリ Node.js版	差異まとめ
1	ジオメトリ情報⇒空間ID情報	<ul style="list-style-type: none"> ジオメトリ情報⇒空間ID ジオメトリ情報配列⇒空間ID配列 ジオメトリ情報配列(直径込み)⇒空間ID配列 3Dデータ⇒空間ID配列 	<ul style="list-style-type: none"> ジオメトリ情報⇒空間ID ZFKYI情報⇒空間ID 	<ul style="list-style-type: none"> (Python版)配列での参照 (Python版)直径を含む経路の参照 (Python版)3Dデータでの参照 (Node.js版)ZFKYIでの参照
2	空間ID情報⇒ジオメトリ情報	<ul style="list-style-type: none"> 空間ID⇒ジオメトリ情報(頂点の8座標配列) 	<ul style="list-style-type: none"> 空間ID⇒ジオメトリ情報(中心点の座標) 	<ul style="list-style-type: none"> 中心点を取得か、頂点座標を取得か
3	周辺空間IDの取得	<ul style="list-style-type: none"> 空間ID⇒周辺の空間ID配列(8or8or20個) ジオメトリ情報⇒周辺の空間ID配列(8or8or20個) 緯度・経度・高さ方向の指定数値分の移動先空間ID 	<ul style="list-style-type: none"> 空間ID⇒周辺の空間ID配列(20個) 空間ID⇒上方向の空間ID 空間ID⇒下方向の空間ID 空間ID⇒東方向の空間ID 空間ID⇒西方向の空間ID 空間ID⇒南方向の空間ID 空間ID⇒北方向の空間ID 	<ul style="list-style-type: none"> 周辺IDの取得範囲の差異 (Python版)ジオメトリ情報から直接参照
4	ズームレベルの変更		<ul style="list-style-type: none"> 空間ID⇒親ズームレベルの空間ID 空間ID⇒子ズームレベルの空間ID配列 	(Node.js版)分解能を変更した空間IDの参照
5	判定機能		<ul style="list-style-type: none"> 空間ID情報/ジオメトリ情報⇒空間IDにジオメトリ情報が含まれるか否かのbool値 	(Node.js版)衝突判定等を想定した機能
6	その他	<ul style="list-style-type: none"> 座標変換(地理座標⇒投影座標) 		(Python版)座標変換機能
	(参考)それぞれのライブラリの設計思想	<ul style="list-style-type: none"> オブジェクトは変数を持たない関数の集合 関数の返り値は基本的に空間IDもしくはIDの配列 	<ul style="list-style-type: none"> オブジェクトは変数と関数を持つクラス 関数の返り値は基本的にオブジェクト(空間情報を変数として保持) 上記で「空間ID」と表記しているものは実際は「空間IDオブジェクト」 	

3.2.5. インポート機能

インポート機能は、データ提供者が提供する各種のデータを空間ID対応システムに取り込む機能であり、図 3-9 に示す空間ID対応システムの構成要素として位置づけられている。ここでは、インポート機能の概念図を図 3-16 に示す。

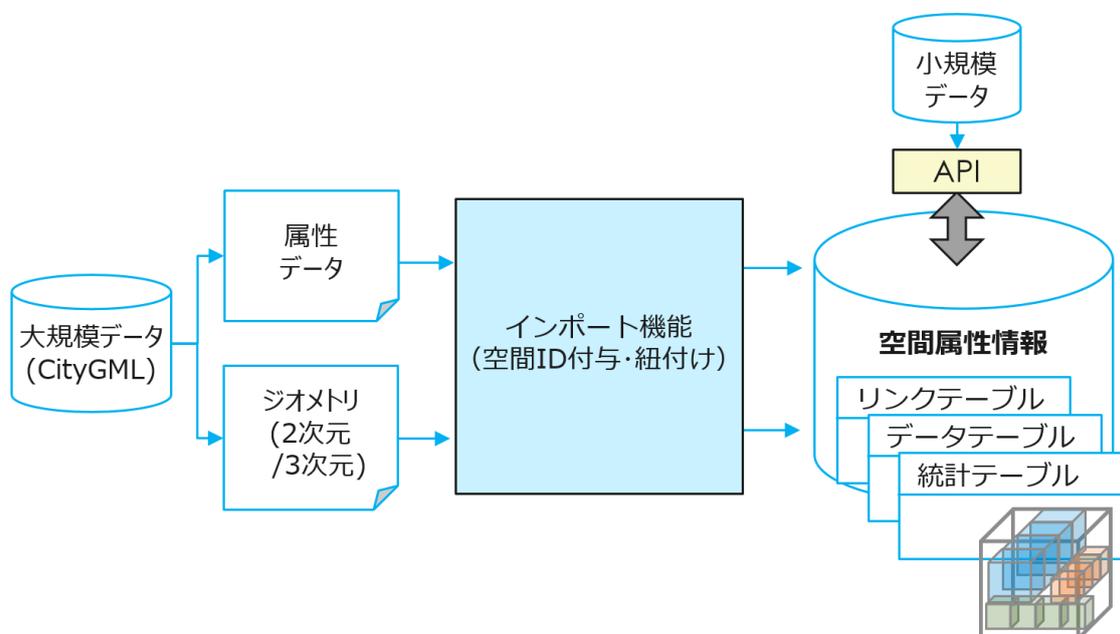


図 3-16 インポート機能の概念図

インポート機能は、データの初期整備等の大規模なデータを一括で読み込む機能を指す。小規模なデータを高頻度で読み込む場合には、事業者が個別に定義する API を用いる。API については、図 3-9 に示す空間ID対応システムの構成要素になっている。

空間属性情報の DB スキーマについては、ユースケース・事業者ごとに整備するデータの内容や仕様によって異なるため、本ガイドラインでは共通仕様化しない。そのため、インポート機能は、競争領域として事業者及び開発者において自由の実装することを想定している。本節では開発の生産性を高める観点から、大規模データの 1 つである 3D 都市モデルを例に、インポート機能の考え方を説明する。

本ガイドラインにおけるインポート機能の役割を以下に示す。

- ・インポート機能とは、データ提供者が提供するデータを取り込み、ジオメトリに対する地物 ID 及び重畳される空間ボクセルに対応する空間 ID を紐付ける機能である。
- ・図 3-9 で示される通り、API を介した機能実装でも、同様の機能を実現できる。
- ・空間 ID の付与方法には、共通ライブラリを使用する方法と、データ提供者が提供するツールを使用する方法がある。
- ・インポート機能の出力は、空間属性情報内のリンクテーブル(地物 ID 及び空間 ID のペアリスト)、データテーブル(ソースデータが保有する属性情報)、又は統計テーブル(空間 ID の単位で集計した属性情報)の 3 種類のテーブルとなる。

空間属性情報の DB スキーマの概念図の例を図 3-17 に示す。

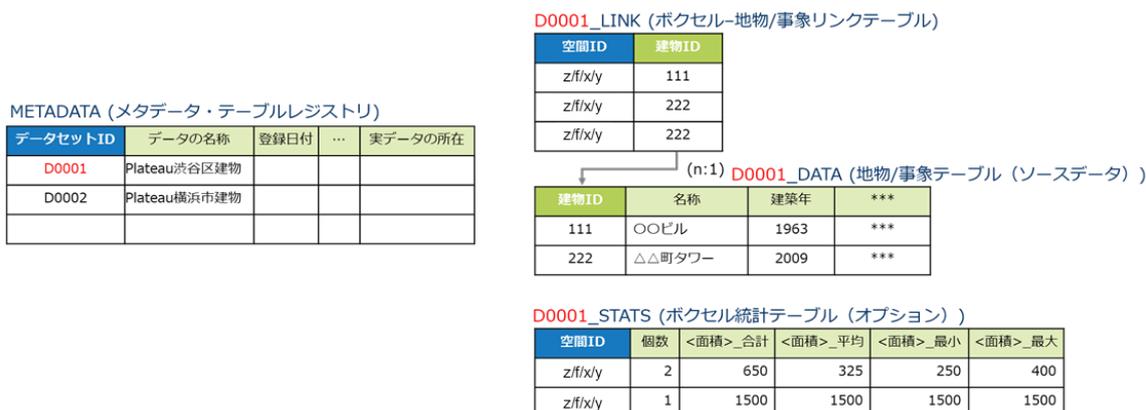


図 3-17 DB スキーマの概念図例

(1) ジオメトリの種類と空間 ID の関係

空間 ID 対応システムにインポートされるデータのジオメトリには、点、線、面及び立体の 4 種類ある。各ジオメトリと空間 ID (空間ボクセル) の関係を表 3-4 に示す。

表 3-4 ジオメトリの種類と空間 ID の関係

ジオメトリの種類	点	線	面	立体
ジオメトリと空間 ID (空間ボクセル) の関係				
	ズームレベルを指定し、点を包含する空間 ID を生成。	ズームレベルを指定し、線を包含する空間 ID を生成。	ズームレベルを指定し、面を含む空間 ID を生成。	ズームレベルを指定し、立体の表面部と内部 (空洞部) に空間 ID を生成。

(2) 3D 都市モデル (CityGML) を例にしたインポート機能の解説

大規模データの 1 つである Project PLATEAU¹¹ の 3D 都市モデル (建築物) を例に、インポート機能の考え方を説明する。

Project PLATEAU の 3D 都市モデルは、国際標準化団体の OGC (Open Geospatial Consortium) が規定する「CityGML 2.0」という国際標準に準拠した「3D 都市モデル標準製品仕様書」によって標準化されている¹²。この 3D 都市モデルから空間 ID を自動生成するツールが、「PLATEAU のための空間 ID 生成ツール (PLATEAU-generator-for-spatialid)」として Project PLATEAU の GitHub 上に公開されている¹³。

この「PLATEAU のための空間 ID 生成ツール」は、次の 2 つの方法で 3D 都市モデルに空間 ID を付与する。3D 都市モデルへの空間 ID の付与方法を表 3-5 に示す。

表 3-5 3D 都市モデルへの空間 ID の付与方法

空間 ID の付与方法		説明
1	CityGML ファイルへ直接付与	CityGML ファイル (データセット) 内に記録される地物インスタンス (地物 ID) ごとに「空間 ID」を記録。
2	外部ファイル (CSV 形式) に付与	CityGML ファイルの肥大化を避ける方法として、地物 ID (gml:id) と「空間 ID」のペアリストを外部ファイルに記録。

空間 ID が直接付与された CityGML (XML) ファイルの例を図 3-18 示す。図中の例示は、XML に直接、空間 ID が付与されている様子を示したものであり、赤枠で囲った箇所が建築物に付与された空間 ID を表している。また、地物 ID-空間 ID のペアリスト CSV ファイルの例を図 3-19 に示す。

¹¹ PLATEAU [プラトー] <https://www.mlit.go.jp/plateau/>

¹² 3D 都市モデル標準製品仕様書 <https://www.mlit.go.jp/plateaudoocument/>

¹³ PLATEAU のための空間 ID 生成ツール <https://github.com/Project-PLATEAU/PLATEAU-generator-for-spatialid>

```

<core:CityModel xmlns:bldg="http://www.opengis.net/citygml/building/2.0" ...>
  <core:cityObjectMember>
    <bldg:Building gml:id="bldg_b90c4f61-79b3-4cff-bcb7-226ab456c6c2">
      <core:creationDate>2023-03-22</core:creationDate>
      <bldg:usage codeSpace=" .././codelists/Building_usage.xml">411</bldg:usage>
      ...
      <uro:buildingSpatialIDAttribute>
        <uro:BuildingSpatialIDAttribute>
          <uro:maxZoomLevel>22</uro:maxZoomLevel>
          <uro:merge>true</uro:merge>
          <uro:spatialID>22/7/1599930/433631</uro:spatialID>
          <uro:spatialID>22/6/1599929/433632</uro:spatialID>
          <uro:spatialID>22/7/1599929/433632</uro:spatialID>
          <uro:spatialID>22/6/1599930/433631</uro:spatialID>
          <uro:spatialID>21/3/799964/216815</uro:spatialID>
        </uro:BuildingSpatialIDAttribute>
      </uro:buildingSpatialIDAttribute>
    </bldg:Building>
  </core:cityObjectMember>
  <uro:externalReferenceOfSpatialID>
    <uro:fileLocation>spatialid/52372215_bldg_6697_zl22_merged.csv</uro:fileLocation>
  </uro:externalReferenceOfSpatialID>
</core:CityModel>

```

3D 都市モデル(建築物)の例

→ 建築物に付与された
空間 ID

図 3-18 空間 ID が直接付与された CityGML (XML) ファイルの例

```

PLATEAU_3D-Spatial-ID_CSV,0100,23,1
gml_id,spatial_id
bldg_b90c4f61-79b3-4cff-bcb7-226ab456c6c2,22/7/1599930/433631
bldg_b90c4f61-79b3-4cff-bcb7-226ab456c6c2,22/6/1599929/433632
bldg_b90c4f61-79b3-4cff-bcb7-226ab456c6c2,22/7/1599929/433632
bldg_b90c4f61-79b3-4cff-bcb7-226ab456c6c2,22/6/1599930/433631
bldg_b90c4f61-79b3-4cff-bcb7-226ab456c6c2,21/3/799964/216815
bldg_f0707e77-9065-4dd6-ac07-603846eea840,22/9/3697091/1663493
bldg_f0707e77-9065-4dd6-ac07-603846eea840,22/10/3697091/1663493
bldg_f0707e77-9065-4dd6-ac07-603846eea840,22/9/3697090/1663493
...

```

先頭行: ヘッダ情報
2 行目: 列名
3 行目以降: 地物 ID と
空間 ID のペアリスト

図 3-19 地物 ID-空間 ID のペアリスト CSV ファイルの例¹⁴

空間 ID 対応システムへ 3D 都市モデル (建築物) の空間属性情報をインポートする方法には、以下の 2 パターンが考えられる。

- リンクテーブルだけを構築し、空間 ID 生成ツールから出力される CSV を使用することで、簡単にリンクテーブルの生成が可能となる。
- CSV (地物 ID と空間 ID のペアリスト) もしくは、CityGML ファイル (XML 直接埋め込み) 内の空間 ID からリンクテーブルを作成し、空間 ID に紐づく地物 ID をキーに CityGML から属性を取得して登録する。

¹⁴ PLATEAU のための空間 ID 生成ツール 技術検証レポート
https://www.mlit.go.jp/plateau/file/libraries/doc/plateau_tech_doc_0055_ver01.pdf

CityGML のデータを空間 ID に変換する際には、空間 ID 対応システムにおいて共通ライブラリ（ジオメトリから空間 ID に変換する機能）を用いて空間 ID を付与する場合及びデータ提供者において事前に空間 ID を付与する場合（想定ケース①）が存在する。

また、データ提供者において事前に空間 ID を付与する場合には、属性情報は扱うことができないが、地物 ID 及び空間 ID のペアリストの CSV ファイルを用いるケース（想定ケース②）、空間 ID が付与されている XML ファイルを用いるケース（想定ケース③）に分けられる。想定ケースごとのインポート機能の概要と特徴を表 3-6 に示す。

表 3-6 想定ケースごとのインポート機能の概要と特徴

#	区分	想定ケース	インポート機能の概要	特徴
①	インポート機能の中で空間 ID を付与する場合	インポート機能の中で共通ライブラリを使用することもでき、地物 ID に空間 ID を紐付けて使用するケース	データからジオメトリを読み込んで、インポート機能の内部でジオメトリを空間 ID に変換する。その際に共通ライブラリを利用することもできる。	<ul style="list-style-type: none"> 共通ライブラリを使って変換できる（ジオメトリ-空間 ID への変換） 現状では、CityGML を属性情報付きのアプリケーション用ファイルに変換してから使用するため外部アプリ（FME Desktop 等）が必要になる。 インポートする際にジオメトリ-空間 ID 変換を行うため、待ち時間が長くなる。
② ③ 共通	データ提供者が提供するデータに空間 ID を事前に付与する場合	データ（ジオメトリ）を事前に変換し、空間 ID を紐付けて使用するケース	データからジオメトリを読み込んで、事前に空間 ID を付与したデータファイルをインポートする際に読み込む。	<ul style="list-style-type: none"> バックグラウンド処理が可能になり、インポート時に待ち時間が少なくなる。 複数時の事業者が同じ変換する場合に事前に変換しておくことで一度処理することでインポートすることが可能になる。
②		属性情報を含まない地物 ID-空間 ID のペアリスト (CSV) を読み込んで使用するケース	データセット単位 (図郭単位) の CSV ファイルを読み込むことで全ての地物 ID に対応した空間 ID を読み込む。	<ul style="list-style-type: none"> インポートの際に対応づけが不要となる (読み込むだけで利用できる) 地物 ID ごとの分離は可能である。 属性情報を参照できない。
③		CityGML (XML) を参照して属性情報付きで地物情報を読み込んで使用するケース	属性情報もあらかじめ付与されているところに空間 ID を付与した XML を読み込む。	<ul style="list-style-type: none"> リンク情報付きの XML、もしくは、単一の空間 ID が付与された XML ファイルを読み込むことで必要な属性情報が得られる。 詳細度の高いズームレベルを対象とする場合はファイルサイズが膨大になる。

(3) インポート機能の中で空間 ID を付与する場合

インポート機能は、競争領域として事業者及び開発者において自由に実装することを想定しているが、開発生産性を高める観点から、本ガイドラインにおいて具体例を示すに当たっては、共通ライブラリにおける 2 次元又は 3 次元の点、線、面、立体のジオメトリから空間 ID に変換する機能を用いることを前提に置く。共通ライブラリにおける同機能は、ズームレベルを指定した上でリンクテーブル(ジオメトリと空間 ID の紐付け)を出力することが主な機能になっている。共通ライブラリで全ての処理を行えるわけではないので、その場合には事業者及び開発者において必要な機能を実装する必要がある。

インポート機能の中で空間 ID を付与する場合のインポート機能を図 3-20 に示す。

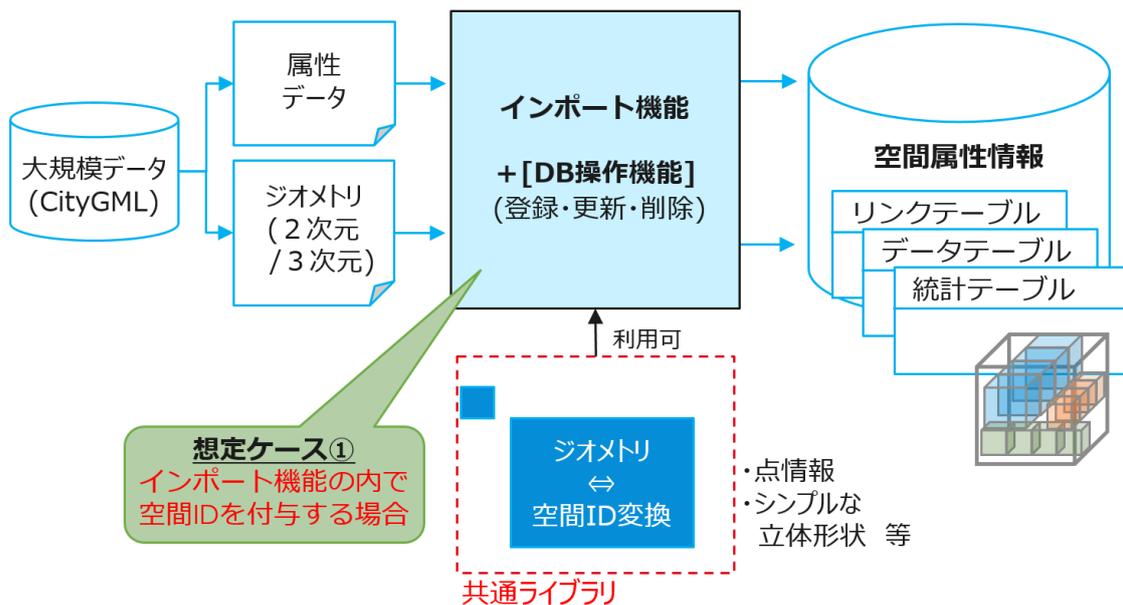


図 3-20 インポート機能の中で空間 ID を付与する場合のインポート機能

(4) データ提供者が提供するデータに空間 ID を事前に付与する場合

データ提供者が提供しているデータコンバータを用いることで、CityGML から直接に地物 ID と空間 ID のペアリストを CSV ファイルで生成することや、XML ファイルにおいて CSV ファイルへのリンクを付すること、CityGML ファイルに空間 ID を直接に付与することができる。これらのファイルをインポートする機能(想定ケース②～③)の概念を図 3-21 に示す。

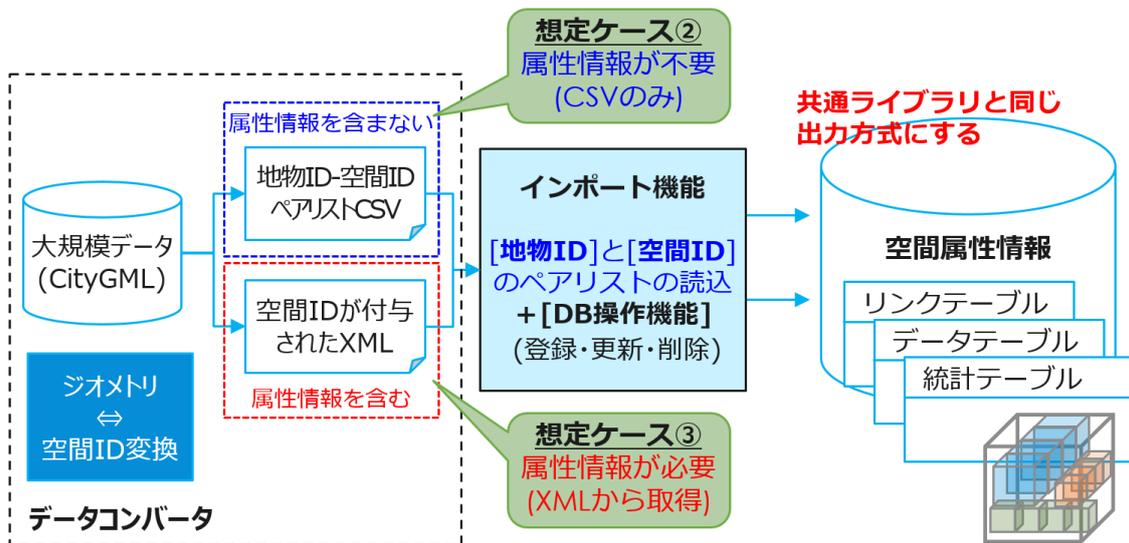


図 3-21 データ提供者が提供するデータに空間 ID を事前に付与する場合のインポート機能

想定ケース②は、属性情報を使用しないケースであり、ドローン等のユースケースにおいて種別を問わずに障害物を回避するようなケースが該当する。この場合、属性情報を含まない地物 ID と空間 ID のペアリストが列挙された CSV ファイルをインポートすることで、空間 ID に対応した空間ボクセルの情報を得ることができる。

想定ケース③は属性情報を必要とするケースであり、様々なユースケースでの利用が想定される。このような場合、CityGML ファイルに属性情報が格納されているため、リンクテーブルを作成した後、データテーブルを作成するために、ユースケースで使用する属性情報を CityGML ファイルから取得する必要がある。

(5) 空間属性情報の操作

空間属性情報の操作を含むデータのインポート機能については、本ガイドラインに記載している共通ライブラリ(3.2.2 節)や、あらかじめデータコンバータで変換したファイルを読み込むことで、各事業者が個別に実装することを想定している。

インポート機能の出力である空間属性情報はリンクテーブル、データテーブル、統計テーブルで構成される。表 3-7 には、その定義、取得・生成方法、出力項目の例を示す。

表 3-7 空間属性情報の構成要素

構成要素	リンクテーブル	データテーブル	統計テーブル
定義	地物 ID と空間 ID のペアリスト	地物 ID に紐づく属性情報(事象)のリスト	空間 ID に紐づく集計した統計情報のリスト
取得・生成方法	共通ライブラリ又はデータコンバータでジオメトリを変換して空間 ID を生成する。	データが保有する属性情報を取得してリストを生成する。	空間 ID の単位で属性情報を集計して統計情報としてのリストを生成する。
出力項目の例	地物 ID、空間 ID	地物 ID、名称、建築年等	空間 ID、個数、面積合計値、面積平均値、最大値、最小値等

空間属性情報の操作(登録・更新・削除)の各機能は、共通ライブラリ(3.2.2 節)と出力形式を合わせることで、リンクテーブル、データテーブル、統計テーブルに対するインポート機能を共通化することが可能になる。

3.3. ユースケース事例

3.3.1. 空間 ID の活用可能性

異なる基準に基づいた空間情報であっても一意に位置を特定できる空間 ID に多様なデータを紐付けることで、空、地上、地下、屋内、海といった空間で、空間属性情報を活用した既存サービスの高度化や新たなサービスの創出が期待される。例えば、空ではドローンを用いたインフラ設備点検や物資の輸送が、地上では自動運転車の安全な運行が、屋内ではロボットを活用した建設現場支援や AR による屋内ナビゲーションが、地下では地下埋設物の可視化や掘削工事の高度化等のユースケースが創出される。

図 3-22 に空間 ID に紐付けるデータの例を空、地上、地下の利用空間別に示す。

利用空間	空	地上 (屋内を含む)	地下
空間IDへ紐づけるデータ (例)	地物情報	地物情報	地形情報
	地形情報	地形情報	電力に係る地下埋設物
	規制情報 (飛行禁止/制限空域)	規制情報 (通行不可エリア)	ガスに係る地下埋設物
	緊急規制情報	BIM情報	上下水道に係る地下埋設物
	電波情報	POI情報 (建物入り口 等)	通信に係る地下埋設物
	気象情報	人流情報	占用事業者名
	飛行計画情報	災害情報	掘削後地面形状情報
	人流情報	気象情報	
	各種イベント情報	路側センサ情報	
		車両情報	

図 3-22 利用空間別に空間 ID に紐付けるデータの例

3.3.2. ユースケース事例

ユースケース事例および各事例の利用空間（空、地上、地下）を表 3-8 に示す。各事例の詳細は APPENDIX-3 に示す。

表 3-8 ユースケース事例一覧（表内の項番は APPENDIX-3 内の項番を記載）

ユースケース事例	利用空間
1.1 地下埋設物管理ユースケース	地下
1.2 複雑な建物内での移動・輸送ユースケース	地上（屋内）
1.3 時間軸を考慮した災害情報の統合・提供ユースケース	地上
1.4 BIM・建物データ等を連携させた CPS ユースケース （ロボット運行最適化、空間の広告価値評価）	地上（屋内）
2.1 地下埋設物管理ユースケースにおける機能の更新・追加	地下
2.2 対象設備への地上設備の追加に関する実証	地上・地下
2.3 遠隔操作のユースケースの追加に関する実証	地下
2.4 災害対応ユースケース	地下
3.1 自動車ユースケース	地上
4.1 物資輸送ユースケース	空
4.2 送電設備点検ユースケース	空
4.3 橋梁点検ユースケース	空
4.4 UTM 間連携に関する技術検証	空
4.5 損害保険に係るリスクアセスメント等に関する技術検証	空
4.6 点検・防災・有人機連携ユースケース	空
4.7 空間情報基盤の有用性に関する技術検証	空
4.8 点検ユースケース	空
4.9 信号認証技術による Spoofing 脅威の低減に関する技術検証	空
4.10 教育ユースケース	地上
4.11 インフラ管理ユースケース	地上
4.12 建設現場アプリユースケース	地上（屋内）
4.13 建設現場アプリに関する技術検証	地上（屋内）
4.14 建設ロボットユースケース	地上（屋内）
4.15 建設ロボットに関する技術検証	地上（屋内）
4.16 空間エンタメユースケース	空・地上
4.17 空間エンタメに関する技術検証	空・地上

3.4. 空間 ID 対応システムの運用

システム間の相互運用性や開発者の開発生産性を向上するために、ユースケース横断で頻出する共通的な機能についてライブラリを整備し、OSS として公開する。また、空間 ID 対応システムにおいては、利用者が「同一機能は同一の手順で空間属性情報を生成・参照・更新・削除」することを実現することを目的として、利用頻度が高いと考えられる機能について、API 仕様を策定する。こうした共通ライブラリ及びインターフェース仕様を格納するリポジトリを作成して運用する。

(1) リポジトリの概要

政府等において開発したプログラムである共通ライブラリ及びインターフェース仕様を独立行政法人情報処理推進機構（以下、「IPA」という。）が運営する GitHub 上のリポジトリに公表する。GitHub 上の共通ライブラリ及びインターフェース仕様については、利用者からの要望及び提案等を踏まえ、必要に応じて更新する。

(2) 公開中のリポジトリ

GitHub で試行的に公開中のリポジトリ URL は以下の通りである。

<https://github.com/ouranos-gex>

4. 今後の展望

4.1. ロードマップ

空間IDによる4次元時空間情報の利活用については、2021年12月から検討を開始して、2022年7月に報告書を取りまとめた。2022年度からは、開発・実証を実施しながら、アーキテクチャ及び技術仕様を具体化してきた。これらを踏まえて、2025年3月に本ガイドラインを公表するに至った。今後は、開発・実証の成果に基づく社会実装の拡大に向けた普及施策の検討とともに、国際標準化によるインセンティブ・エンフォースメント設定やOSS運用を含めた持続的な推進体制を構築する。

4.2. 普及及び標準化に向けた取組

普及に向けては、2024年度に策定されたデジタルライフライン全国総合整備計画における社会への先行実装の取り組みであるアーリーハーベストプロジェクトのドローン航路、自動運転サービス支援道、インフラ管理DXにおいて、空間情報を扱う際の標準識別子として空間IDが採用されており、これらの取り組みと連携することで基幹ユースケースを整備するとともに、実証事業の成果や先行事例に基づき、多数のユースケースの創出を促していく。デジタルライフライン全国総合整備計画の概要を図4-1に示す。



図4-1 デジタルライフライン全国総合整備計画の概要 ¹⁵

¹⁵ 経済産業省 第2期デジタルライフライン全国総合整備実現会議

https://www.meti.go.jp/policy/mono_info_service/digital_architecture/lifeline.html

標準化に向けては、市場でのシェア拡大によるデファクト標準の獲得と並行して、デジタル標準の獲得を両輪で進めることが重要である。企業や業界を横断したデータの連携・活用を実現するために、技術仕様の標準化による妥当性の担保、ユースケースの標準化による特性や拡張性の訴求、協調領域の OSS 整備、ISO による仕様化でのインセンティブ・エンフォースメントの設定といった政策的な取組と普及・標準化の取組を一体的に実施する必要がある。(図 4-2)

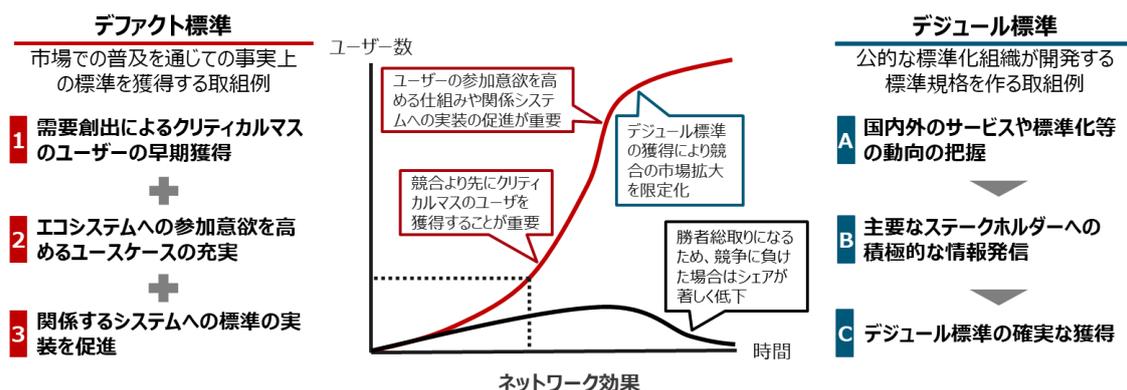


図 4-2 標準化に向けた取組

そのためには、社会に情報発信を行うとともに、ガイドラインや OSS を整備・公表することで、サービス提供者や開発者による実装を促進していく。その際には、ユーザーが抱える課題、要望及び提案をタイムリーに把握して、情報の更新等の対応策を講じるサイクルを定着することが重要であり、空間情報サービスの利用者及び提供者、システム開発者等を中心としたコミュニティの構築等の施策の検討も行う。