

ISSN 1346-7328
国総研資料 第 507 号
平成 21 年 1 月

国土技術政策総合研究所資料

TECHNICAL NOTE of
National Institute for Land and Infrastructure Management

No.507

January 2009

電子地図 / 建設情報連携のための技術資料

金澤文彦・青山憲明・坂森計則・渡辺完弥・今井龍一

Technical Note of Digital Map and Collaboration of
Construction Information

Fumihiko KANAZAWA, Noriaki AOYAMA, Kazunori SAKAMORI,
Kanya WATANABE, Ryuichi IMAI

国土交通省 国土技術政策総合研究所

National Institute for Land and Infrastructure Management
Ministry of Land, Infrastructure, Transport and Tourism, Japan

電子地図 / 建設情報連携のための技術資料

金澤文彦* , 青山憲明* , 坂森計則* , 渡辺完弥* , 今井龍一**

Technical Note of Digital Map and Collaboration of Construction Information

Fumihiko KANAZAWA, Noriaki AOYAMA, Kazunori SAKAMORI,
Kanya WATANABE, Ryuichi IMAI

概要

本研究では、建設情報の連携による業務の効率化の実現を目的に次の技術資料を作成した。

- ・ 建設情報連携ポータル標準インタフェースガイドライン (案)
- ・ 地名辞典の整備・運用ガイドライン (案)
- ・ CAD-GIS 連携の手引き書 (案)

キーワード :

情報連携 , 地名辞典 , CAD , GIS

Synopsis

In this study, we made the next technical note on collaboration of construction information for the promotion of efficiency of operations.

- ・ Guideline for Standard Interface of Portal for Collaboration of Construction Information
- ・ Guideline for maintenance and operation of the gazetteer in public works
- ・ Guideline with regard to convert between CAD and GIS

Key Words :

Collaboration of Information, Gazetteer, CAD, GIS

* 高度情報化研究センター 情報基盤研究室
Information Technology Division, Research Center for Advanced Information Technology

** 日本工営株式会社 (元 高度情報化研究センター 情報基盤研究室)
Nippon Koei Co., Ltd

はじめに

公共発注機関では、CALS/ECの一環として電子納品を推進しており、建設情報の電子データの蓄積および流通による再利用の環境を整備している。また、さまざまなシステムやデータベースが構築され、多種多様で大量の建設情報が分散管理されている。こうした情報化の推進により、電子化された建設情報が徐々に蓄積されてきている。

電子化された建設情報が蓄積されてきている一方、今後の方向性として、分散管理された建設情報を計画・調査～維持管理のライフサイクルを通じて、いかに有効に活用するかに注目が集まってきている。すなわち、建設情報の連携による業務の効率化の実現が求められている。

“分散管理された建設情報の有効活用”を行うために、建設情報の多くは位置情報と関連づけることができることから、位置情報を連携キーとするのが得策である。具体的には、住所や地名などの“地理識別子”と経緯度情報などの“座標”とを関連づけた語彙集である地名辞典を整備することにより、さまざまなアプリケーションでの位置情報の利活用性が向上する。また、蓄積された建設情報を有効活用するためには、情報システムの連携による相互運用性を高めていく必要がある。この方策として、既存のデータベースの構造に大きな変化を加えることなく、標準的なインタフェースを用いて分散環境にある情報システムを連携させることが挙げられる。

一方、公共事業の“ライフサイクルを通じた建設情報の利用”を図るうえで重要となる建設情報として成果図面が挙げられる。業務の効率化を実現するためには、各フェーズにおける図面の用途・要件などの特性を踏まえた建設情報の交換・流通を図ることが望まれる。具体的には、測量成果である“拡張DMデータ形式の図面情報”、設計成果である“CADデータ形式の図面情報”、維持管理で用いる“GISデータ形式の図面情報”の3つを対象にしたデータ交換・流通の環境の整備が必要となる。

本研究では、建設情報の連携による業務の効率化の実現を目的に、“分散管理された建設情報の有効活用”として、「地名辞典の整備・運用ガイドライン(案)」および「建設情報連携ポータル標準インタフェースガイドライン(案)」を作成した。また、“ライフサイクルを通じた建設情報の利用”として、「CAD-GIS連携の手引き書(案)」を作成した。

これらの資料の作成にあたっては、建設情報標準化委員会 電子地図/建設情報連携小委員会(柴崎亮介小委員長)に審議を依頼した。ここに、多大なご協力をいただいた関係者に対して謝意と敬意を表する次第である。

1. 建設情報連携に関する研究の背景、目的

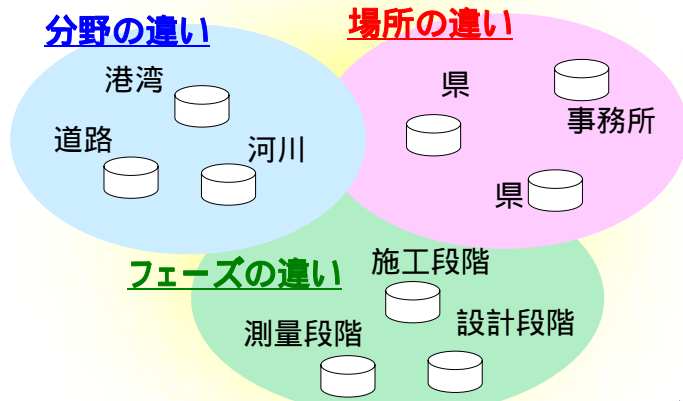
公共発注機関では、CALS/EC (Continuous Acquisition and Life-cycle Support/ Electronic Commerce ; 公共事業支援統合情報システム) の一環として電子納品を推進しており、建設情報の電子データの蓄積・流通による再利用の環境を整備している。また、さまざまなシステムやデータベースが構築され、多種多様で大量の建設情報が分散管理されている。こうした情報化の推進により、電子化された建設情報が徐々に蓄積されてきている。

しかしながら、現状では、電子データは分野ごと、場所ごと、事業フェーズごとにそれぞれ作成、管理されており、分野、場所、事業フェーズを跨いだ情報連携は不十分である。そのため、必要な情報がなかなか見つからない、入手した情報をそのまま利用できない、という問題が発生している。

今後の方向性として、分野ごと、場所ごと、事業フェーズごとに分散管理された建設情報を連携し、電子データの検索の効率化、再利用性の向上を図り、業務の効率化の実現が求められている。

このような背景を受けて、高度情報化研究センター 情報基盤研究室では、建設情報を連携するための制度、仕組み、標準等の研究を行っている。本書では、これまでの研究成果である位置情報をキーとして建設情報を電子地図上で連携するための「地名辞典」、分散管理されているシステムやデータベースを連携するための「建設情報連携ポータル標準インタフェース」、施工フェーズと維持管理フェーズの図面データを連携するための「CAD-GIS 連携」について報告する。

各所に分散している電子データ



建設情報を連携するための制度、仕組み、標準等

- ・位置情報をキーとして建設情報を電子地図上で連携するための「地名辞典」
- ・分散管理されているシステムやデータベースを連携するための「建設情報連携ポータル標準インタフェース」
- ・施工フェーズと維持管理フェーズの図面データを連携するための「CAD-GIS連携」



2. 研究全体と今回策定した基準類の位置づけ

国土交通省では、組織間、事業段階間で公共事業に関する情報の交換、連携を図り、コスト縮減、品質確保、事業執行の効率化を目指して、CALS/ECに取り組んでいる。

CALS/EC 推進の実施計画である CALS/EC アクションプログラム 2005 では、「(1)情報交換」、「(2)情報共有・連携」、「(3)業務プロセス改善」、「(4)技術標準」、「(5)国際交流・連携」に分けて全 18 の目標を設定している。

本研究は、建設情報連携をテーマとしているが、CALS/EC アクションプログラム 2005 の目標のうち、「(2)情報共有・連携」に該当する以下の目標が本研究と密接に関連する。

- 目標-9 . 完成図を利用した管理図の蓄積・更新の迅速化・効率化
- 目標-10 . 維持管理データベース更新の迅速化・効率化
- 目標-11 . GIS 管理図に重ね合わせた施設情報管理の効率化
- 目標-13 . 情報モデルの管理によるシステム間の情報交換・共有・連携の促進

CALS/ECアクションプログラム2005の目標

情報交換	入札契約	目標-1. 入札契約情報の提供方法の工夫による情報収集効率の向上 目標-2. 入札説明書のインターネットを通じた配布による調達手続きの効率化 目標-3. 契約手続きの電子化による調達手続きの効率化	
	電子納品	目標-4. CADデータ交換標準の改良による情報交換の効率化 目標-5. 3次元情報の利用を促進する要領整備による設計・施工管理の高度化	
情報共有・連携	入札契約	目標-6. 入札契約手続に関するシステム間連携による調達手続きの効率化	
	電子納品	目標-7. 地質データの提供による調査分析・施工計画の精度向上 目標-8. 施設情報を提供して技術提案募集によるコスト縮減と品質確保 目標-9. 完成図を利用した管理図の蓄積・更新の迅速化・効率化 目標-10. 維持管理データベース更新の迅速化・効率化 目標-11. GIS管理図に重ね合わせた施設情報管理の効率化 目標-12. 現場からの情報取得による作業の効率化 目標-13. 情報モデルの管理によるシステム間の情報交換・共有・連携の促進	
		共通	目標-14. 取組状況の公開と研修テキストの共有による全国的展開の促進
		業務プロセスの改善	電子納品
工事施工中の情報共有			目標-16. 工事施工中の情報交換・共有の効率化
技術標準		目標-17. 情報共有・連携に向けた必要な標準の整備	
国際交流・連携		目標-18. CADの高度利用へ対応した国際標準機関との連携	

目標-9では「CAD-GIS連携」、目標-10では「建設情報連携ポータル標準インタフェース」、目標-11では「地名辞典」、「建設情報連携ポータル標準インタフェース」、目標-13では「地名辞典」、「建設情報連携ポータル標準インタフェース」の技術をそれぞれ活用することで、効率的なデータ連携、システム連携を実現し、目標を達成することができる。

本研究では、「地名辞典」、「建設情報連携ポータル標準インタフェース」、「CAD-GIS連携」に関する次の標準を作成した。

- 地名辞典の整備・運用ガイドライン（案）
- 建設情報連携ポータル標準インタフェースガイドライン（案）
- CAD-GIS連携の手引き書（案）

目標番号	目標
目標-9 .完成図を利用した管理図の蓄積・更新の迅速化・効率化	完成図（維持・修繕工事を含む）は工事完了後の形状が反映され、工事完成後の測量は最小限になる。 完成図CADデータから更新情報を迅速にGISデータへ反映する。
目標-10 .維持管理データベース更新の迅速化・効率化	設計、工事（維持・修繕工事を含む）で作成している台帳・DBの情報項目を電子納品する。 電子納品情報から関連する台帳・DBに対して一括して更新作業を行う。
目標-11 .GIS管理図に重ね合わせた施設情報管理の効率化	必要な情報の検索及び集約を瞬時に行える。 施設形状のGISデータを整備して、維持管理情報を重ね合わせて表示する。
目標-13 .情報モデルの管理によるシステム間の情報交換・共有・連携の促進	登録簿を参照して、入手したいデータがどのDBで管理しているか所在が明確になり、情報の交換・共有がしやすくなる。 システム統合せずに、情報連携を順次拡張することが可能となる。

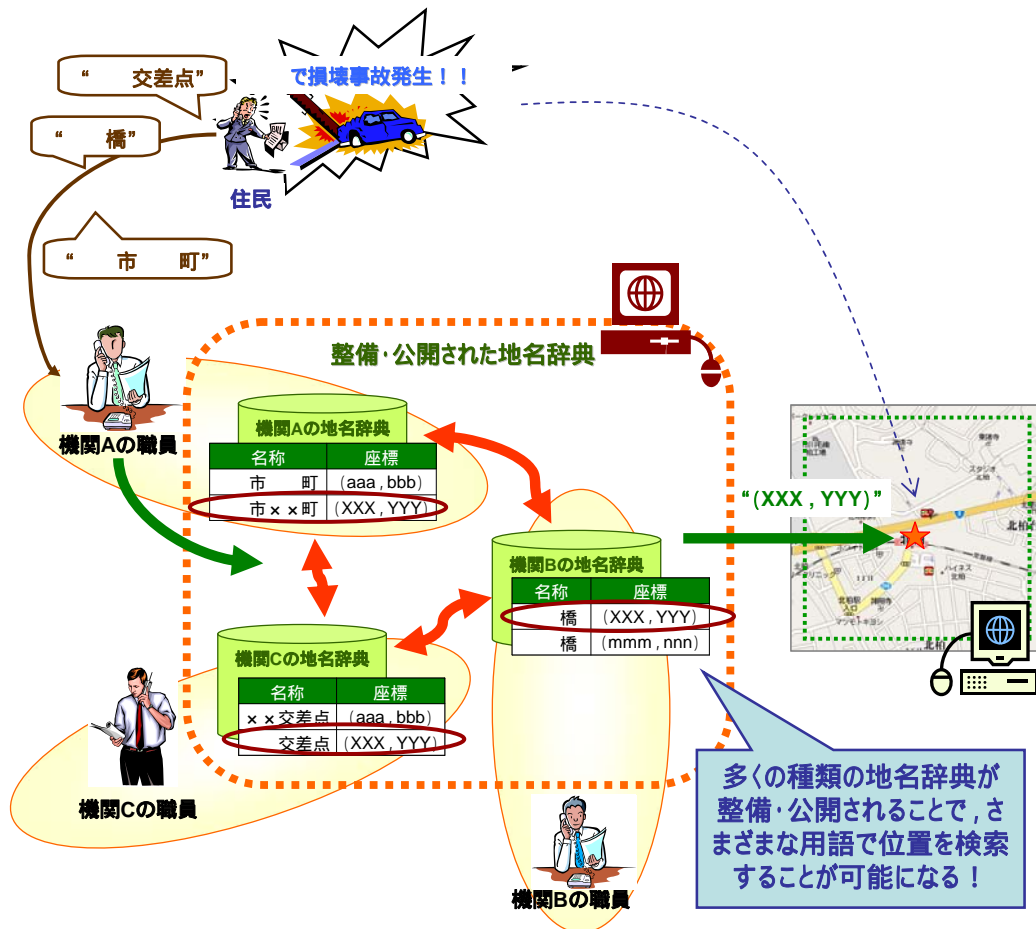
3. 地名辞典の整備・運用ガイドライン(案)の研究概要

建設情報の多くは位置情報と関連づけることができるため、建設情報の検索には位置情報を連携キーとするのが得策である。コンピュータが位置を特定する最も有効な情報は緯度・経度であるが、人が位置を検索するキーとしては利用しにくい。住所や地名、施設名等、普段に位置を識別する情報(地名)をキーとして、建設情報を検索することが一般的である。

地名辞典とは、住所や地名などの“地理識別子”と経緯度情報などの“座標”とを関連づけた語彙集であり、位置情報に関する辞典である。

現在、住所から緯度・経度への変換、郵便番号から住所への変換といった位置情報の変換は、すでにGISなどで実現されているが、変換の方法などについて標準的な共通の決まりは存在しない。今後、地名辞典はさまざまな機関で作成されることが予想されるが、データ構造や整備・更新する際に守るべき運用ルールなどを標準化することで、異なるシステムでの地名辞典の利用や、重複のない効率的な地名辞典の整備が可能となる。

「地名辞典の整備・運用ガイドライン(案)」は、地名辞典の整備・運用の考え方、データ構造、運用ルールを規定した資料であり、地名辞典を整備、作成する方、地名辞典を利用する方及び地名辞典を利用したサービス・アプリケーションを開発する方を対象として作成している。



4. 建設情報連携ポータル標準インタフェースガイドライン(案)の研究

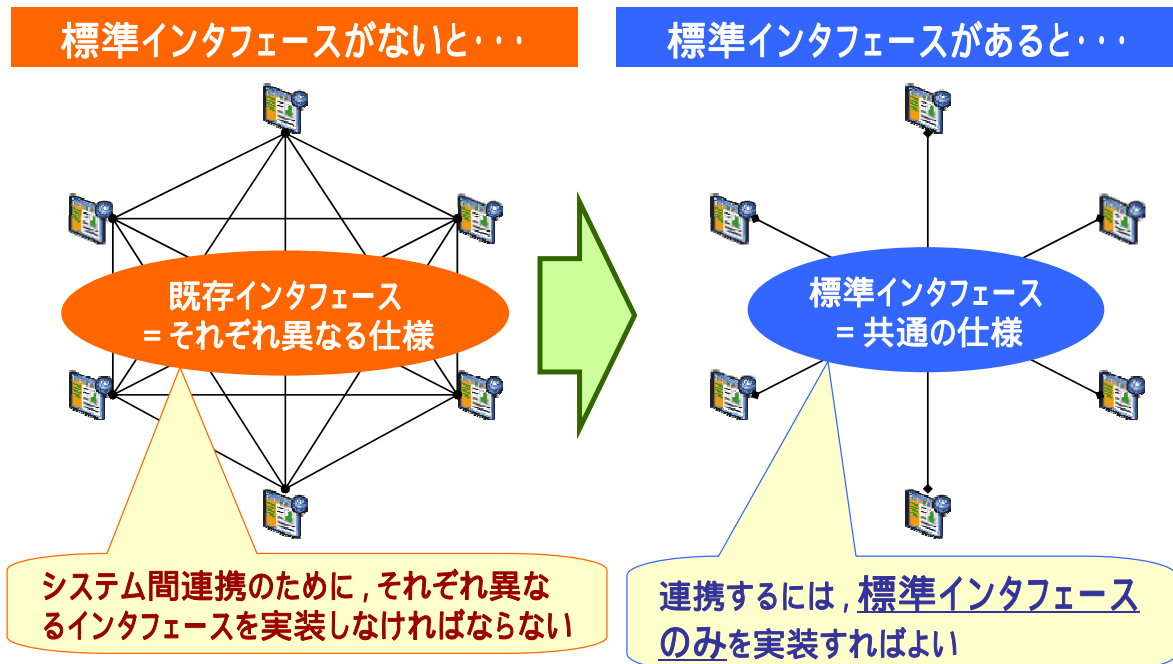
概要

公共事業では、ライフサイクルの各段階の業務を支援するために、さまざまな情報システムが整備・運用されている。これらのシステムの多くは、システム間の連携が困難であり、データの連携ができないだけでなく、利用者にとっても様々なシステムを操作しなければならない不便な状況を作っている。異なるシステム間において円滑な情報共有を実現するために、システム間の情報連携の標準的なルールとして「標準インタフェース」を定め、導入することにより、システムの相互運用性が高まり、分野横断的な情報共有が可能となる。

標準インタフェースの検討に当たっては、先進的に取り組んでいる河川 GIS・河川アプリケーション標準インタフェースガイドライン、減災情報共有プロトコル、地理情報共用 Web システムガイドライン、PI (場所識別子) 仕様書を参考としている。

「建設情報連携ポータル標準インタフェースガイドライン(案)」では、建設情報を取り扱う情報システム間の連携に有効と考えられる標準インタフェースとして、次の 3 種類を定義した。

- 地理情報インタフェース：地理情報の表示や属性表示など，地理情報の制御機能を実現
- メタデータ検索インタフェース：メタデータを利用した情報検索やその結果表示の機能を実現
- 地名辞典インタフェース：地名検索の機能を実現



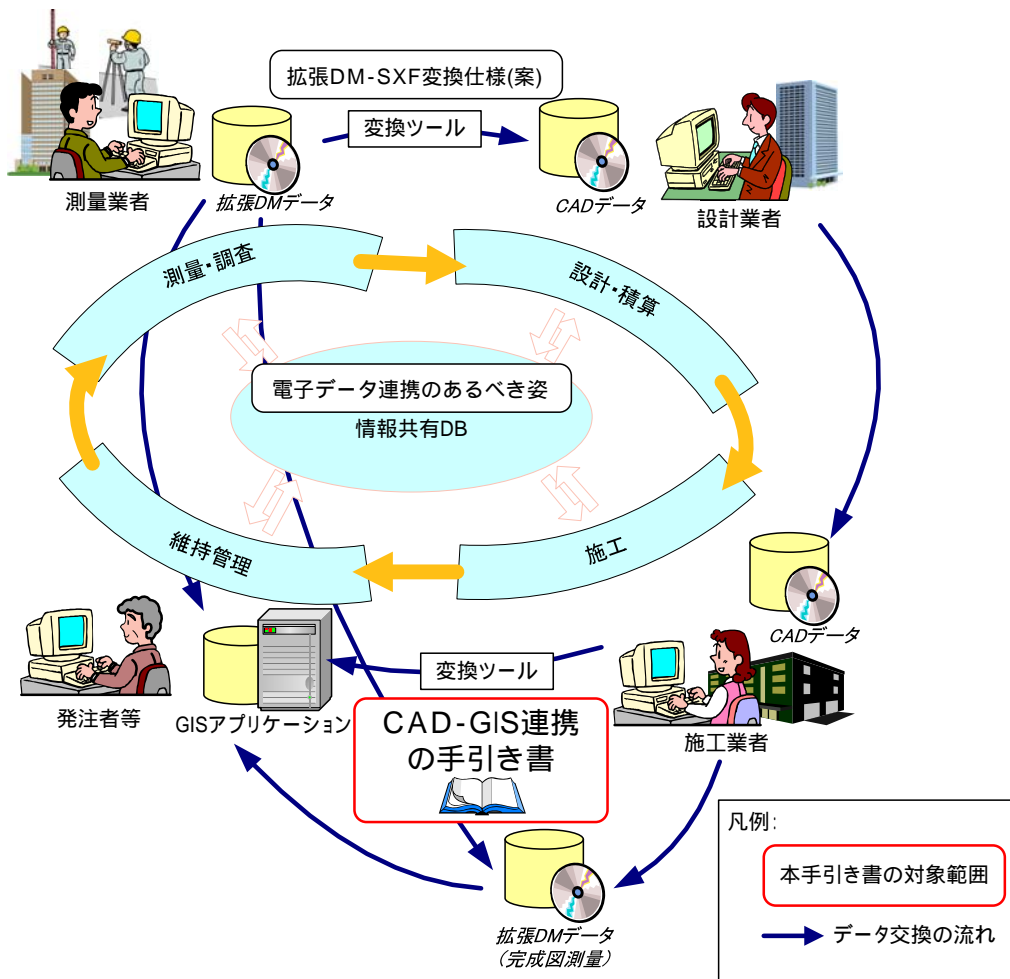
5. CAD-GIS 連携の手引き書(案)の研究概要

建設事業におけるライフサイクル（測量・調査、設計・積算、施工、維持管理）において、測量・調査フェーズでは DM データ、設計・積算、施工フェーズでは CAD データ、維持管理段階では GIS データがそれぞれ利用されている。フェーズ間のデータ交換・連携として、測量・調査と設計フェーズ間の「DM-CAD 連携」、施工と維持管理フェーズ間の「CAD-GIS 連携」が必要となる。

DM-CAD 連携については、「拡張 DM-SXF 変換仕様(案)」が策定され、測量（DM データ）から設計（CAD データ）への図面データの受け渡し場面などで利用されている。本研究では、残された課題として、施工（CAD データ）と維持管理（GIS データ）とを連携する CAD-GIS 連携について検討している。

CAD-GIS 連携の手引き書(案)は、維持管理段階で利用する GIS アプリケーションの GIS データ（基盤地図データ）に、施工段階で電子納品される CAD データを有効利用したいと考えている発注者が、CAD-GIS 間のデータ変換を進めるにあたり必要となる基準類等の作成を検討する際の見方をとりまとめたものである。

CAD-GIS 連携の手引き書(案)に準じて必要な基準類等を作成することにより、CAD データから GIS データへの変換作成の実現を目指すことを目的としている。



建設情報連携のためのポータル標準インタフェースガイドライン(案)
第1版

平成19年10月

国土交通省

はじめに

公共事業では、ライフサイクルの各段階の業務を支援するために、さまざまな情報システムが整備・運用されている。これらのシステムは、それぞれの目的に応じて独立に運用され効果を上げているが、複数の情報システムを連携して利用できれば、さらに大きな効果が得られると期待される。しかしシステム間の連携を図るための適切な管理基準や制度が存在しないために、分野横断的な情報連携が困難なことがある。このため、建設分野において流通するデータの標準化、および分散システムの検索・利用技術の確立は、業界全体の効率化推進に特に大きな効果があると期待される。

情報を有効に利活用するためには、「情報の蓄積と管理」が不可欠である。国土交通省では、直轄事業のすべての業務・工事を対象に電子納品を実施している。また、地方自治体でも電子納品が展開されてきており、様々な建設情報が蓄積されている。今後、蓄積された様々な建設情報を効率よく検索・収集・利活用していくためには、情報システムの基盤整備が特に重要な課題と言える。

蓄積された建設情報を有効活用するには、情報システムの連携による相互運用性を高めていく必要がある。相互運用性を高める方策として、既存のデータベースの構造に大きな変化を加えることなく、何らかの標準的なインタフェースを用いて分散環境にある情報システムを連携させる手法が考えられる。建設情報の分野横断的な連携利用の実現のためには、既存の規格や事例を踏まえた、建設分野全体を対象とした分散システムの連携と、データ検索・利用技術の仕組みを開発することが望まれる。

本書は、建設分野で流通する情報を、効率よく検索・収集・利活用できるようにするため、情報システムの連携による相互運用性を高めるための「標準的なインタフェース」として、インターネット/イントラネット環境で動作するアプリケーションプログラムが実装すべきインタフェースの考え方を示すとともに、インタフェースの仕様を定めている。

本書の策定にあたっては、国土交通省が(財)日本建設情報総合センターに業務を委託し、建設情報標準化委員会 電子地図/建設情報連携小委員会(柴崎亮介小委員長)にご検討・ご審議をいただいた。多大なご協力をいただいた関係各位に深く謝意を表する次第である。

平成 19 年 10 月

国土交通省

<目次>

【本編】

1. ガイドラインの概要.....	1
1-1. 目的.....	1
1-2. 本ガイドライン策定の背景と標準インタフェースの意義.....	2
1-3. 適用範囲.....	5
1-4. 利用者と利用方法.....	6
1-5. 引用規格.....	9
1-5-1. 国際的な標準化団体による標準規格.....	9
1-5-2. 既存の標準インタフェース仕様との関係.....	10
1-6. 用語の定義.....	12
2. 標準インタフェースの考え方.....	15
2-1. 本ガイドラインにおける標準インタフェースの位置づけ.....	15
2-2. 標準インタフェースの定義.....	20
2-3. 標準インタフェースの種類.....	21
2-4. 標準インタフェースの定義方法.....	24
2-5. 関数仕様の考え方.....	25
2-6. 関数仕様の実装方式.....	26
2-7. 標準インタフェースを利用したアプリケーションプログラムの運用形態.....	27
2-8. インタフェース公開・管理の考え方.....	28
2-9. インタフェース接続の考え方.....	30
3. 標準インタフェース関数仕様.....	32
3-1. 関数の一覧.....	33
3-2. 関数の説明.....	35
3-2-1. 共通.....	35
3-2-2. 地理情報に関するインタフェース.....	36
3-2-3. 地名辞典に関するインタフェース.....	38
3-2-4. メタデータ検索に関するインタフェース.....	53
3-2-5. 構造体.....	58
4. 標準インタフェース実装仕様.....	82
4-1. POST/GET 利用のための実装仕様.....	82
4-1-1. 地理情報に関するインタフェース.....	82
4-1-2. 地名辞典に関するインタフェース.....	82
4-1-3. メタデータ検索に関するインタフェース.....	82
4-2. SOAP と WSDL 利用のための実装仕様.....	83

4-2-1.	共通事項.....	83
4-2-2.	各サービスに共通するインタフェース.....	86
4-2-3.	地理情報に関するインタフェース.....	89
4-2-4.	地名辞典（空間参照系管理）に関するインタフェース.....	90
4-2-5.	地名辞典（地名辞典管理）に関するインタフェース.....	100
4-2-6.	メタデータ検索に関するインタフェース.....	110
5.	関数仕様作成に関する取り決め.....	111
5-1.	関数仕様作成に関する取り決め.....	111
5-2.	関数と引数の命名規則.....	116
6.	実装仕様作成に関する取り決め.....	118
6-1.	実装仕様作成に関する取り決め.....	118
7.	標準インタフェースの運用に関する取り決め.....	120
7-1.	インタフェース公開・管理方法.....	120
7-2.	インタフェース接続方式.....	122
8.	標準インタフェース公開・管理仕様.....	123
8-1.	アプリケーションプログラムの内容に関する公開・管理仕様.....	124
8-2.	アプリケーションプログラムで利用できる関数等の技術的な公開・管理仕様.....	127

【附属書】

1. 参考とすべきインタフェース.....	1
1-1. 地物単位で地理情報を取得するインタフェース（参考）	2
1-1-1. 関数仕様.....	2
1-1-2. 詳細情報の入手先	3
1-2. SOAP による地理情報等の標準インタフェース（参考）	4
1-2-1. 関数仕様.....	5
1-2-2. 詳細情報の入手先	10
2. ガイドラインの適用事例.....	11
2-1. システム実装の事例.....	12
2-1-1. 概要.....	12
2-2. 連携システムの詳細.....	14
2-2-2. 実装の手順（例）	17
2-3. インタフェース策定の事例.....	23
2-3-1. 河川 GIS・河川アプリケーション標準インタフェースガイドラインにお ける事例	23
2-3-2. 電子納品保管管理システムにおける事例	25

1. ガイドラインの概要

1-1. 目的

「建設情報連携のための標準インタフェースガイドライン」(以下、「本ガイドライン」という)は、インターネット/イントラネット環境で動作するアプリケーションプログラムが実装すべきインタフェースの考え方を示すとともに、インタフェースの仕様/取り決めについて定める。

なお本ガイドラインでは、各節の要点を枠内に記述し、本文を枠外に記述する。

【解説】

本ガイドラインでは、国及び地方公共団体などの建設情報を作成、利用する各機関・各部署で採用・開発する情報システムを相互に連携するための考え方と方法論を示している。方法論については、「仕様」と「取り決め」として取りまとめている。本ガイドラインに記載している内容は、概ね以下のとおりである。

- インタフェースの考え方
- インタフェースを利用したシステムの開発と運用
- インタフェースに関わる仕様
- インタフェースに関わる取り決め

1-2. 本ガイドライン策定の背景と標準インターフェースの意義

現在稼働中のアプリケーションプログラムの多くは、システム間の連携が困難であり、データの連携ができないだけでなく、利用者にとっても様々なシステムを操作しなければならない不便な状況を作っている。そこで、異なるアプリケーションプログラム間において円滑な情報共有を実現するために、アプリケーションプログラム間の情報連携の標準的なルールとして「標準インターフェース」を策定し、これを導入することとした。標準インターフェースの導入により、アプリケーションプログラムの相互運用性が高まり、分野横断的な情報共有が可能となる。

【解説】

現在稼働中のアプリケーションプログラムの多くは、開発者の違い、開発言語の違い、プログラムの処理方法の違い、データ仕様の違いにより、システム間の連携が困難であり、データの連携ができないだけでなく、利用者にとっても様々なシステムを操作しなければならない不便な状況を作っている（図 1-1）。

例えば、国道事務所における業務を例にとると、ある道路の盛土区間で亀裂等の変状が見つかった場合、当該区間の状況は道路管理システムで、地質等の状況は地盤情報データベースから、試験等の竣工時の状況は電子納品保管管理システムから、それぞれ検索して入手する必要がある（図 1-2）。

データの取得に関する処理方法を標準化し、それを標準インターフェースとして公開することで、道路管理システムを通じて地盤情報データベースに対して当該区間のボーリング柱状図を要求したり、電子納品保管管理システムに対して当該区間の竣工図書を要求したりすることが可能となる（図 1-3）。これにより、今まで困難であったデータの入手・利用が容易になるだけでなく、新規のシステム開発の場合にすでにあるシステムやデータを利用することが可能となり、データ整備やシステム開発のコストの縮減につながる効果が見込まれる。

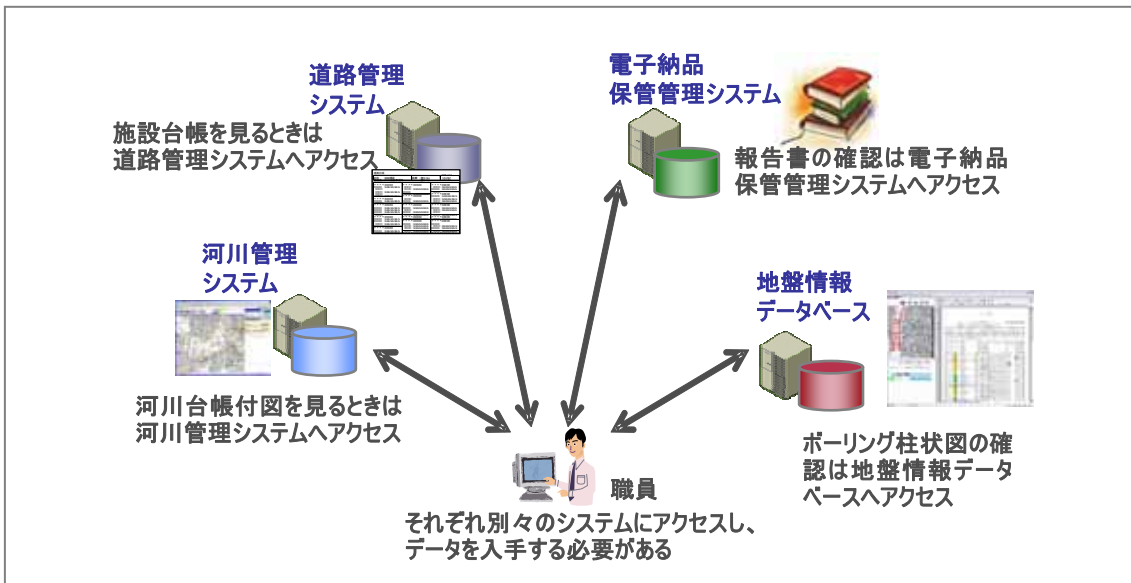


図 1-1 現在のシステム利用状況

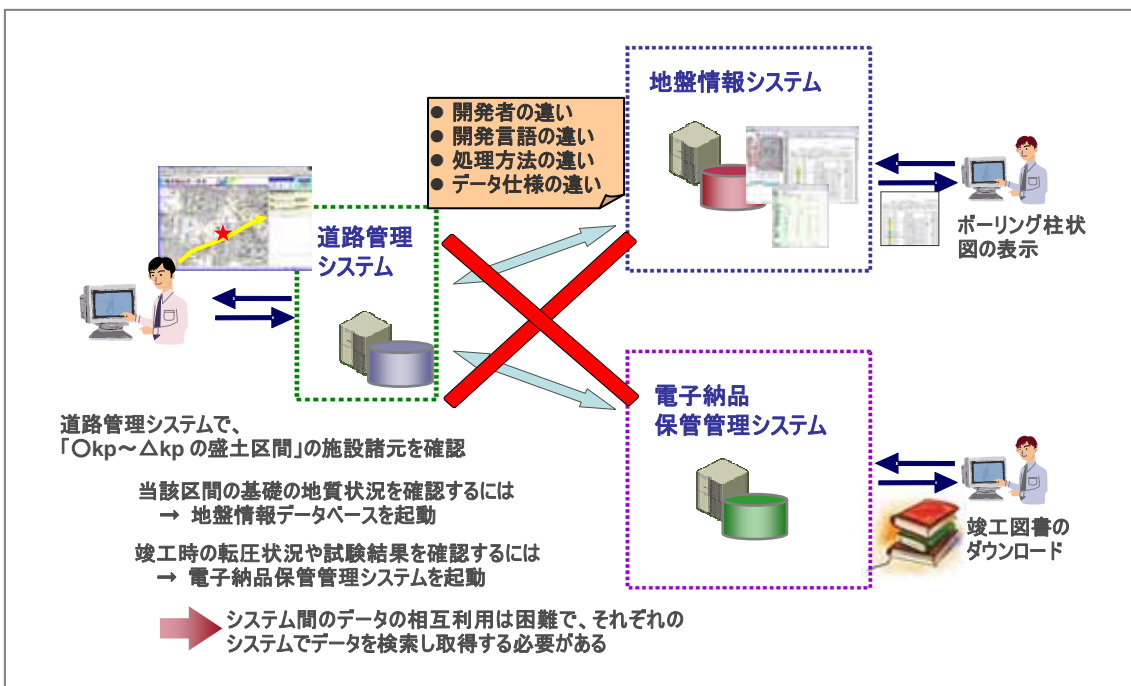


図 1-2 現在のシステム利用の例

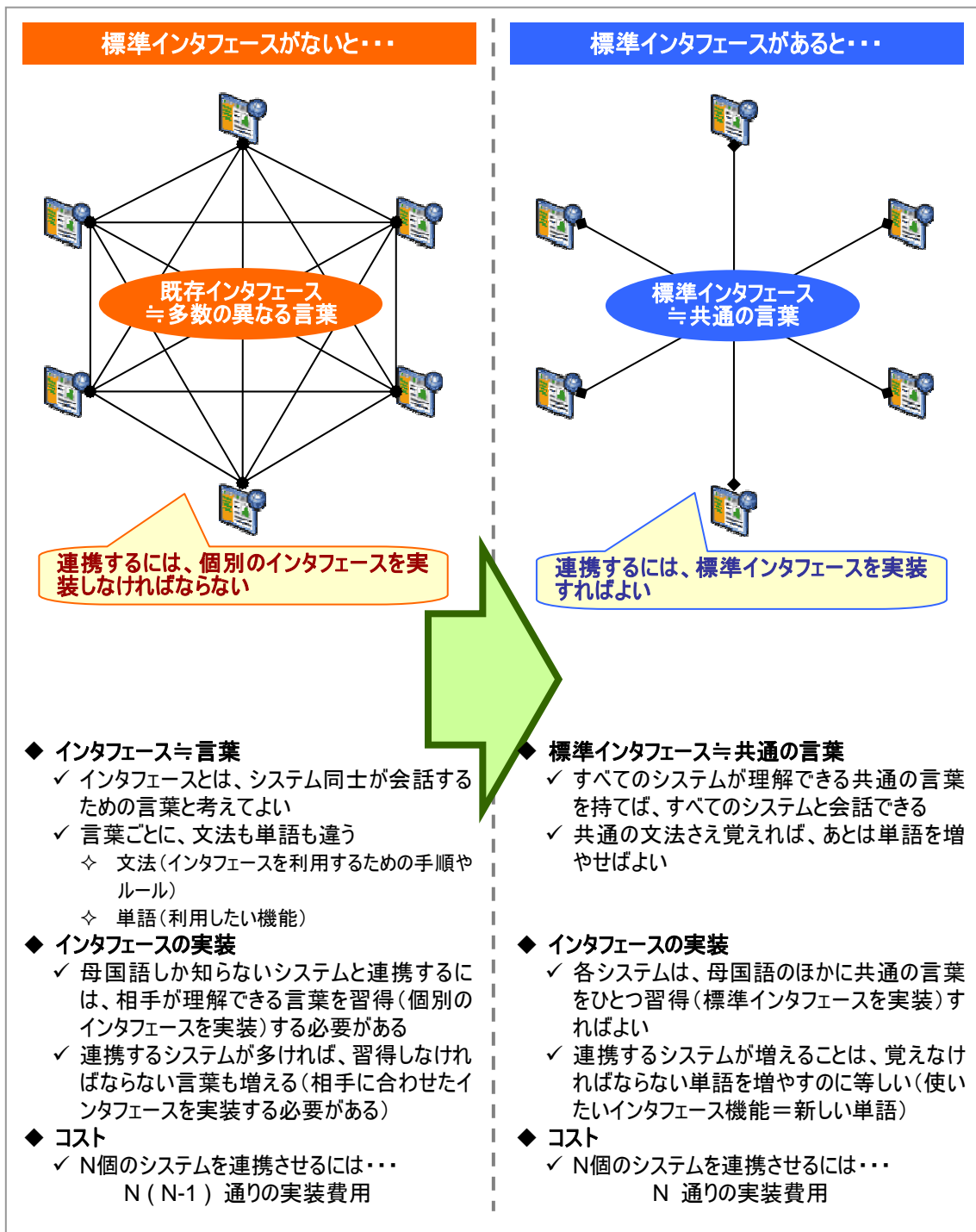


図 1-3 標準インタフェースによる連携のイメージと効果

1-3. 適用範囲

本ガイドラインは、建設情報を取り扱うアプリケーションプログラムの開発及び保守・運用管理に適用する。このうち、データの共用性を考慮し、インターネット/イントラネット環境で動作するアプリケーションプログラムを対象とする。

なお、上記アプリケーションプログラムには、新規に開発するもの、既存のもの(ただし、機能追加や改修の予定があるもの)の両方を含む。

【解説】

インターネット/イントラネット環境で動作するアプリケーションプログラムは、多くの利用者を対象としたものであり、共用性の高いものである。本ガイドラインは、このような共用性の高いアプリケーションプログラムのうち、「建設情報」を取り扱うものに対して適用する。本ガイドラインでは、アプリケーションプログラムを開発・保守・運用管理するにあたっての標準的なルールを定めている。

1-4. 利用者と利用方法

本ガイドラインの利用者は、以下のとおりとする。

- システム開発の企画者
- システム開発の発注者
- システム開発者
- システム管理者

各利用者は、情報システムの企画～運用の各フェーズにおいて、本ガイドラインを参照し、情報システムの管理において自らが担う役割を全うするものとする。

【解説】

本ガイドラインの利用者と利用方法を、表 1-1 の通り定義する。

表 1-1 本ガイドラインの利用者

No.	利用者	役割
1	システム開発の企画者 新規にシステム開発の企画・調整をする立場にある者 稼働中のシステムに対して、システムの機能追加やリニューアルを検討する立場にある者	システム開発の企画者は、ガイドラインの概要や位置づけを把握し、標準インタフェースの考え方を理解した上で、本ガイドラインに従ったシステム開発を企画・検討する
2	システム開発の発注者 システム開発を発注する際の仕様書を作成する立場にある者	システム開発の発注者は、標準インタフェースの考え方を理解し、標準インタフェースの関数仕様を理解した上で、本ガイドラインに従ったシステム開発の発注仕様書を作成する
3	システム開発者 システムを開発する立場にある者(外部委託企業等を含む)	システム開発者は、標準インタフェースの関数仕様を理解した上で、標準インタフェース実装仕様に従ったシステム開発を行う
—	システム利用者 システムを利用する者	本ガイドラインの対象外
4	システム管理者 システムを運用・管理する立場にある者	システム管理者は、標準インタフェースの普及・運用に関する考え方にしたがって、標準インタフェースに対応したシステムの保守・管理・運用を行う

本ガイドラインの利用者は、システム運用の各フェーズにおいて、システムに対して何らかの影響を及ぼす役割を持つ者を想定しており、所謂「システムの利用者」は対象外とする。表 1-1 の内容を図示すると、図 1-4 のようになる。

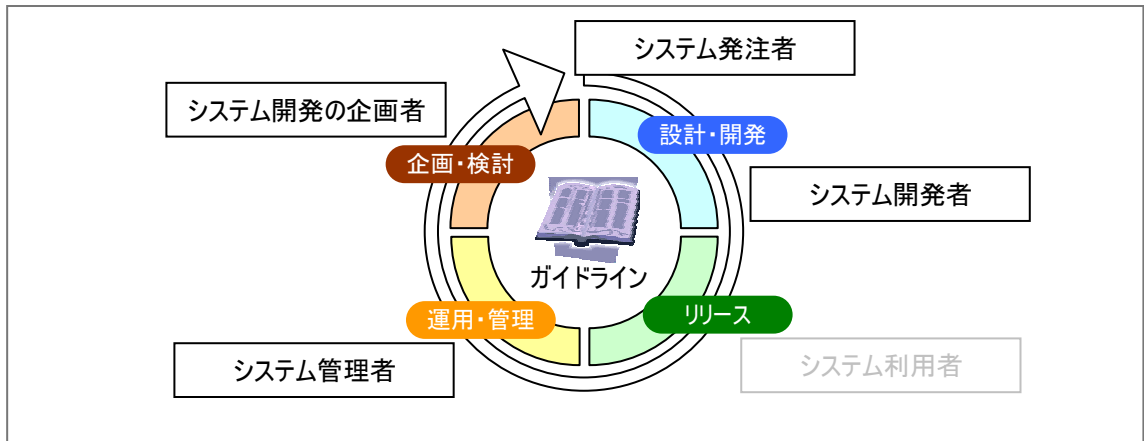


図 1-4 システム運用の各フェーズにおける本ガイドラインの利用者

それぞれの利用対象者が参照すべき本ガイドラインの内容は、表 1-2 に示すとおりである。

表 1-2 利用対象者が参照する本ガイドラインの項目

本ガイドラインの目次	企画者	発注者	開発者	管理者
1.ガイドラインの概要・位置づけ				
1.1.目的	○	○	○	○
1.2.本ガイドライン策定の背景と標準インタフェースの意義	○	○	○	○
1.3.適用範囲	○	○	○	○
1.4.利用者と利用方法	○	○	○	○
1.5.引用規格	○	○	○	○
1.6.用語の定義	○	○	○	○
2.標準インタフェースの考え方				
2.1.本ガイドラインにおける標準インタフェースの位置づけ	○	○	○	○
2.2.標準インタフェースの定義	○	○	○	
2.3.標準インタフェースの種類		○	○	
2.4.標準インタフェースの定義方法		○	○	
2.5.関数仕様の考え方		○	○	
2.6.関数仕様の実装方式		○	○	
2.7.標準インタフェースを利用したアプリケーションプログラムの運用形態		○	○	○
2.8.インタフェース公開・管理の考え方		○	○	○
2.9.インタフェース接続の考え方		○	○	○

本ガイドラインの目次		企画者	発注者	開発者	管理者
3.標準インタフェース関数仕様					
3.1.関数の一覧			○		
3.2.関数の説明			○		
4.標準インタフェース実装仕様					
4.1.POST/GET 利用のための実装仕様			○		
4.2.SOAPとWSDL 利用のための実装仕様			○		
5.関数仕様作成に関する取り決め					
5.1.関数仕様作成に関する取り決め			○		
5.2.関数と引数の命名規則			○		
6.実装仕様作成に関する取り決め					
6.1.実装仕様作成に関する取り決め		○	○	○	
7.標準インタフェースの運用に関する取り決め					
7.1.インタフェース公開・管理方法					○
7.2.インタフェース接続方式					○
8.標準インタフェース公開・管理仕様					
8.1.アプリケーションプログラムの内容に関する公開・管理仕様					○
8.2.アプリケーションプログラムで利用できる関数等の技術的な公開・管理仕様					○
附属書					
1.参考とすべきインタフェース					
1.1.地物単位で地理情報を取得するインタフェース(参考)	○	○	○		
1.2.SOAPによる地理情報等の標準インタフェース(参考)	○	○	○		
2.ガイドラインの適用事例					
2.1.システム実装の事例	○	○	○		
2.2.インタフェース策定の事例	○	○			

1-5. 引用規格

本ガイドラインは、以下に掲げる標準規格を引用する。これにより、本ガイドラインを構成する。

1-5-1. 国際的な標準化団体による標準規格

【W3C : World Wide Web Consortium】

- Simple Object Access Protocol (SOAP) 1.1, W3C Note 08 May 2000
- Web Services Architecture, W3C Working Group Note 11 February 2004
- Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001

【ISO : International Organization for Standardization】

- ISO19128 Web Map Server interface(WMS)

【OGC : Open Geospaial Consortium, Inc.】

- OGC Web Services Common Specification Version: 1.1.0 with Corrigendum 1 2007-02-09
- OGC Filter Encoding Implementation Specification Version: 1.1.0 2005-05-03
- OGC Catalogue Services Specification Version: 2.0.0 with Corregendum 2005-05-20

【JIS : 日本工業規格】

- JIS X7112 地理識別子による空間参照

【その他】

- Basic Profile Version 1.0, 16 April 2004, The Web Services-Interoperability Organization (WS-I)

1-5-2. 既存の標準インタフェース仕様との関係

本ガイドラインを作成するにあたり、以下に挙げる既存の標準インタフェースを大いに参考とした。また、いくつかの既存標準インタフェースからは、その一部を引用している。具体的な引用箇所を、表 1-3 に示す。

- 減災情報共有プロトコル Ver. 1.00.028s 平成 18 年 9 月
独立行政法人 防災科学技術研究所/独立行政法人 産業技術総合研究所
- 平成 18 年度情報家電活用基盤整備事業（時空間情報利活用システム推進事業）
「PI の標準化と実装検証」 平成 19 年 2 月
財団法人 日本情報処理開発協会 データベース振興センター
- 地理情報共用 Web システム標準インタフェースガイドライン 第 0.03 版 平成 13 年 3 月
国土交通省国土計画局
- 河川 GIS・河川アプリケーション標準インタフェースガイドライン 第 1.0 版
2006 年 6 月 27 日
国土交通省河川局

表 1-3 既存の標準インタフェースからの引用箇所

本ガイドラインの目次	他の既存インタフェースからの引用			
	減災	PI	WMS	河川
1. ガイドラインの概要・位置づけ				
1.1. 目的				
1.2. 本ガイドライン策定の背景と標準インタフェースの意義				
1.3. 適用範囲				
1.4. 利用者と利用方法				
1.5. 引用規格				
1.6. 用語の定義				○
2. 標準インタフェースの考え方				
2.1. 本ガイドラインにおける標準インタフェースの位置づけ				
2.2. 標準インタフェースの定義				○
2.3. 標準インタフェースの種類				
2.4. 標準インタフェースの定義方法				○
2.5. 関数仕様の考え方				○
2.6. 関数仕様の実装方式				○

本ガイドラインの目次	他の既存インタフェースからの引用			
	減災	PI	WMS	河川
2.7.標準インタフェースを利用したアプリケーションプログラムの運用形態				○
2.8.インタフェース公開・管理の考え方				○
2.9.インタフェース接続の考え方				○
3.標準インタフェース関数仕様				
3.1.関数の一覧		△	△	△
3.2.関数の説明		△	△	△
4.標準インタフェース実装仕様				
4.1.POST/GET 利用のための実装仕様		△	△	△
4.2.SOAPとWSDL 利用のための実装仕様		△	△	△
5.関数仕様作成に関する取り決め				
5.1.関数仕様作成に関する取り決め				○
5.2.関数と引数の命名規則				○
6.実装仕様作成に関する取り決め				
6.1.実装仕様作成に関する取り決め				○
7.標準インタフェースの運用に関する取り決め				
7.1.インタフェース公開・管理方法				○
7.2.インタフェース接続方式				○
8.標準インタフェース公開・管理仕様				
8.1.アプリケーションプログラムの内容に関する公開・管理仕様				○
8.2.アプリケーションプログラムで利用できる関数等の技術的な公開・管理仕様				○
附属書				
1.参考とすべきインタフェース				
1.1.地物単位で地理情報を取得するインタフェース(参考)	△			
1.2.SOAPによる地理情報等の標準インタフェース(参考)				△
2.ガイドラインの適用事例				
2.1.システム実装の事例				
2.2.インタフェース策定の事例				△
減災： 減災情報共有プロトコル PI： PIの標準化と実装検証 WMS： 地理情報共有 Web システム標準インタフェースガイドライン 河川： 河川 GIS・河川アプリケーション標準インタフェースガイドライン (○…引用、△…一部引用)				

1-6. 用語の定義

本ガイドラインで利用する用語を、以下のとおり定義する。

(i) アプリケーションプログラム

プログラム言語の要素を論理的に組み合わせ、PC システムで機械、プロセスなどを制御するために必要なプログラムの集合体
(JIS B3500 プログラマブルコントローラ用語)

(ii) インターネット【Internet】

複数のネットワークを相互に接続して構成されるネットワーク。ここでは、イントラネットと区別し、TCP/IP 技術を基礎とした世界中を接続するネットワークの意味で用いる。

(改訂電子情報通信用語辞典：電子情報通信学会編)

(iii) イン트라ネット【Intranet】

インターネット向けに開発された技術やインフラを LAN に適用したシステム。
(情報処理技術者用語辞典：日経 BP 社)

(iv) インタフェース【Interface】

ハードウェアやソフトウェアを相互に接続するための仕様（規格）や取り決め（規約）。

(情報処理技術者用語辞典：日経 BP 社 を基に定義)

(v) Web サービス【Web Service】

インターネットプロトコルをプロトコルとするネットワーク上で、共通の操作で機械間の相互作用ができるように構成されたソフトウェア・システム

(Web Services Architecture :W3C Working Group Note、11 February 2004 を基に定義)

(vi) サービス【service】

アプリケーションなどが所定のインタフェースを通して命令を受けることにより実行される機能で、他から区別して一まとまりものとして認識されるもの

(ISO19119 Geographic information – Services を基に定義)

(vii) クライアント

二つ以上の情報処理機器をネットワークでつないで利用するもののうち、処理を依頼する側。ユーザ（利用者）が操作する端末。

（改訂電子情報通信用語辞典：電子情報通信学会編）

(viii) サーバ

二つ以上の情報処理機器をネットワークでつないで利用するもののうち、共通に利用される専門処理機能をもつもの。

（改訂電子情報通信用語辞典：電子情報通信学会編）

(ix) POST/GET

HTTP プロトコルで規定されているリクエストのメソッドの一部。

(x) 略語

GIS	Geographic Information System (地理情報システム ¹⁾)
HTML	Hypertext Markup Language (ハイパテキストマーク付け言語 ²⁾)
HTTP	Hypertext Transfer Protocol (ハイパテキスト転送プロトコル ³⁾)
ISO	International Organization for Standardization (国際標準化機構 ⁴⁾)
NPO	Nonprofit Organization (行政・企業とは別に社会的活動をする非営利の民間組織 ⁵⁾)
OGC	Open Geospatial Consortium、 Inc.
SOAP	Simple Object Access Protocol (XML によるデータ交換のための通信プロトコル ⁶⁾)
W3C	World Wide Web Consortium (1994 年 10 月に設立された World Wide Web に利用される技術の標準化を進める国際的な非営利団体 ⁶⁾)
WFS	Web Feature Service
WMS	Web Map Server、 Web Map Service
WSDL	Web Service Description Language (Web サービスのサービス仕様を記述 ⁶⁾)
XML	Extensible Markup Language (拡張可能なマーク付け言語 ⁷⁾)

注 ¹⁾ : 地理情報科学事典 : 地理情報システム学会編

注 ²⁾ : JIS X 4156:2000

注 ³⁾ : TS X 0085:2004

注 ⁴⁾ : 改訂電子情報通信用語辞典 : 電子情報通信学会編

注 ⁵⁾ : 広辞苑第五版 : 岩波書店

注 ⁶⁾ : 情報処理技術者用語辞典 : 日経 BP 社

注 ⁷⁾ : JIS X 4159:2002

2. 標準インタフェースの考え方

2-1. 本ガイドラインにおける標準インタフェースの位置づけ

本ガイドラインでいう「標準インタフェース」とは、異なるアプリケーションプログラム間において情報を共有し、新たなサービスを作成するために利用するためのインタフェースである。

本ガイドラインにおける標準インタフェースは、既存の標準インタフェースにおいて策定されている内容から共用性・汎用性の高いものを抽出し、再利用可能となるよう再構成したものと、既存の標準インタフェースでは未検討であるが、多くの分野で応用可能なものを定義している。

【解説】

(1) 既存インタフェースについて

標準インタフェースについては、本ガイドライン以外にも、(独)防災科学技術研究所、(財)日本情報処理開発協会データベース振興センター、国土交通省国土計画局、国土交通省河川局等においても、各機関が関係する分野を対象とした検討が実施されている。

- 減災情報共有プロトコル Ver. 1.00.028s 平成 18 年 9 月
独立行政法人 防災科学技術研究所/独立行政法人 産業技術総合研究所
- 平成 18 年度情報家電活用基盤整備事業（時空間情報利活用システム推進事業）
「PI の標準化と実装検証」 平成 19 年 2 月
財団法人 日本情報処理開発協会 データベース振興センター
- 地理情報共用 Web システム標準インタフェースガイドライン 第 0.03 版 平成 13 年 3 月
国土交通省国土計画局
- 河川 GIS・河川アプリケーション標準インタフェースガイドライン 第 1.0 版
2006 年 6 月 27 日
国土交通省河川局

それぞれの標準インタフェースについて、特徴、概要、適用範囲 / 分野等のカテゴリで整理すると、表 2-1 のようになる。

表 2-1 各標準インタフェースの関係

整理項目	河川 I/F	減災 I/F	地理 Web	PI 標準
適用範囲(対象)	河川分野のデータ取得	防災(減災)分野の情報共有	地理情報全般	PI(Place Identifier: 場所を特定する名前)
記述内容	技術仕様+運用指針等のガイドライン	技術仕様	技術仕様+運用指針等のガイドライン	技術仕様
採用している引用規格(地理情報)	WMS (ISO19128)	WFS (OGC 実装仕様)	WMS (ISO19128)	ISO19112
採用している引用規格(プロトコル)	HTTP、SOAP	SOAP	HTTP	HTTP、SOAP
他の分野への適用可能性	<ul style="list-style-type: none"> ✓ 地理情報に関する I/F ✓ プロトコルの仕様 ✓ 規格引用の考え方 ✓ 拡張の考え方 	<ul style="list-style-type: none"> ✓ 地理情報に関する I/F ✓ プロトコルの仕様 ✓ 規格引用の考え方 	<ul style="list-style-type: none"> ✓ 地理情報に関する I/F ✓ プロトコルの仕様 ✓ 規格引用の考え方 	<ul style="list-style-type: none"> ✓ 地理情報に関する I/F ✓ プロトコルの仕様 ✓ 規格引用の考え方

【表題凡例】	
河川 I/F:	河川 GIS・河川アプリケーション標準インタフェースガイドライン
減災 I/F:	減災情報共有プロトコル
地理 Web:	地理情報共用 Web システムガイドライン
PI 標準:	PI の標準化と実証検証
技術仕様:	通信プロトコルや関数仕様等、インタフェースを利用したアプリケーションを作成するのに必要な技術文書という位置づけ(主に開発者向け)
運用指針:	インタフェースを実際に普及・利用するための考え方や、運用上のルール等の施策までを含んだ文書という位置づけ

(2) 本ガイドラインにおける標準インタフェースの位置づけ

本ガイドラインにおいて策定した標準インタフェースは、図 2-1 に示すように、既存の標準インタフェースのうち、他分野に適用可能な部分に関しては、それを包含し、既存の標準インタフェースでは規定されていない部分に関しては、新たに検討・策定したものである。したがって、既存のインタフェース仕様で定められている内容は尊重し、今後、現在定められていない分野（道路、港湾、下水道など）においても同様の考え方で適用できるようなインタフェース仕様となっている。

		河川 I/F	地理 Web	減災 I/F	PI 標準	etc...							建設 I/F
汎仕様	地理情報	○	○	○	○	→							●
	メタデータ検索	※ここでいう「メタデータ検索」とは、クリアリングハウスを利用した横断検索を指す											◎
	地名辞典				○	→							●
技術仕様		○	○	○	○	→							●
考え方等	運用ルール	○	○	→	→	→							●
	拡張ルール	○	→	→	→	→							●
	開発手法	○	○	○	○	→							●

既存の標準インタフェースは、適用範囲や策定期間が異なること等の理由から、相互に連携して策定されていない状況である。
記述内容には、重複・類似する箇所も存在する。

本件における標準インタフェースは建設情報という広い対象を想定しているため、既存のインタフェースを包括した汎用的な記述となる。

【凡例】

<p>河川 I/F: 河川 GIS・河川アプリケーション標準インタフェースガイドライン</p> <p>減災 I/F: 減災情報共有プロトコル</p> <p>地理 Web: 地理情報共用 Web システムガイドライン</p> <p>PI 標準: PI の標準化と実証検証</p> <p>建設 I/F: 本件における標準インタフェース</p>	<p>既存 I/F で定義されている事項に関しては、整理・統合が可能なものは整理し、既存 I/F のものをベースに策定。 ○⇒●</p> <p>既存 I/F で定義されていないものは、新規に策定。 ⇒◎</p>
--	---

図 2-1 既存の標準インタフェースと本件の標準インタフェースの策定内容

(3) 本ガイドラインにおける標準インターフェースと今後の利用方法

本ガイドラインで定める標準インターフェースは、以下の3種類である。

- 1) 標準インターフェース導入の考え方等、分野を問わない普遍的な概念
- 2) 共用性の高い技術仕様
- 3) 汎用性の高いインターフェース仕様

これらは、様々な建設情報を連携して利用する上で必要な事項を定めたものであるが、個々の内容についてはすでに一般的な地理空間情報を利用するために定められたものである。

本ガイドラインにおいて汎用的なインターフェースを策定したことにより、上記に示したような異なるインターフェース間の重複は解消される。各応用分野では図 2-2 に示すように、本ガイドラインにおいて策定した標準インターフェースをベースとし、特定分野の利用に特化したインターフェースを定めることができる。

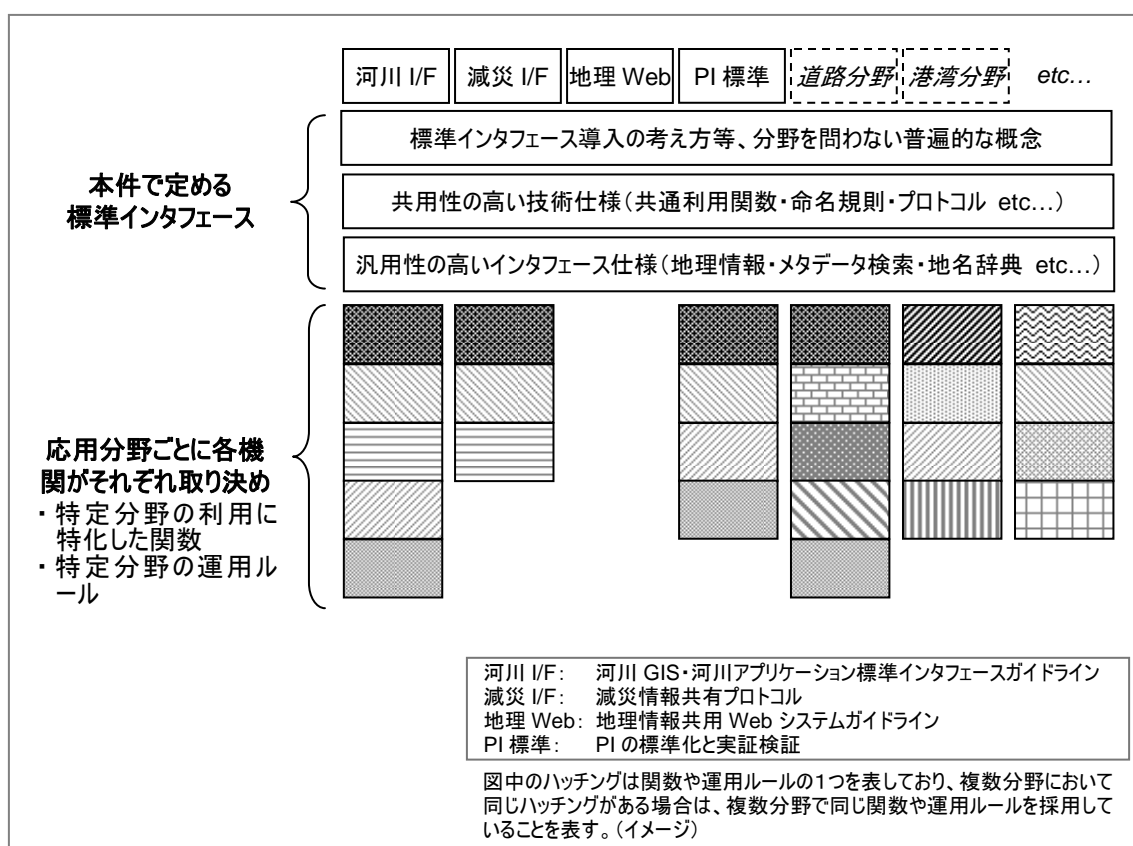


図 2-2 本ガイドラインの標準インターフェースと他の標準インターフェースの関係

(4) 本ガイドラインの位置づけと各分野のメリット

以上のように、本ガイドラインにおいて定義する標準インタフェースとは、特定の分野ではなく、建設情報全般のシステム開発において汎用的に利用可能なものである。以下に、本ガイドラインの位置づけ、および各分野に本ガイドラインを適用した際のメリットを整理した。

【本ガイドラインで定める標準インタフェース】

- 地理情報、メタデータ検索、地名辞典等、汎用性の高い部分を策定
- 標準インタフェースの考え方等、適用分野に左右されない普遍的な概念を定義

【他の分野における標準インタフェース】

- 本ガイドラインの標準インタフェースをベースとし、応用分野ごとに必要な部分を策定
- 河川、減災等の既存分野だけでなく、道路、港湾等の分野へも展開予定

【各分野におけるメリット】

- 各分野のシステムから、他分野のシステムの持つ情報を収集・利用することが可能となる
 - ◇ 他分野の最新の情報を直接入手でき、データの受け渡しが効率化する
- 標準インタフェースの運用について、各々の分野に特化した関数や運用ルールのみを管理(維持・更新)すればよく、それぞれの管理対象を明確化できる
 - ◇ 分野横断的な部分のメンテナンスにかかる費用・時間が削減できる

2-2. 標準インターフェースの定義

本ガイドラインで適用するインターフェースとは、インターネット/イントラネット環境で動作するアプリケーションプログラム間を相互接続するための仕様と取り決めを指す。

【解説】

本ガイドラインで適用するインターフェースは、関数と実装方式だけを取り決めるものではなく、それを運用するときのルールも含む。本ガイドラインでは、表 2-2 で “ ” 印をつけた項目について、仕様と取り決めを定める。

表 2-2 本ガイドラインで定めるインターフェースに関する仕様と取り決め

内容		仕様	取り決め
インターフェース	関数について	○	○
	実装方式について	○	○
	接続方式について	△	○
運用	公開・管理について	○	○

○:本ガイドラインで定める項目

△:すでにある仕様・取り決めを採用する項目

本ガイドラインでは、標準インターフェース作成に関する取り決めとして、「関数仕様作成に関する取り決め」、「実装仕様作成に関する取り決め」、「接続方式に関する取り決め」、「運用に関する取り決め」を定める。それらの取り決めに基づき、「関数仕様」「実装仕様」「公開・管理仕様」を定める。

2-3. 標準インターフェースの種類

本件における標準インターフェースは、その機能によって以下の3種類を定義する。

- (1) 地理情報に関するインターフェース
- (2) 地名辞典に関するインターフェース
- (3) メタデータ検索に関するインターフェース

【解説】

本件における標準インターフェースは、その機能によって以下に大分できる。それぞれの種類について概要を説明する

(1) 地理情報に関するインターフェース

ここでいう「地理情報に関するインターフェース」とは、建設情報を共有する場面を想定したとき、地理情報に関するサービスとして以下の機能を実現するために必要なインターフェースである。

表 2-3 地理情報に関するインターフェース

No.	機能	具体例
1	地図を表示する(座標指定)	地名辞典で地名を検索し、地名に該当する座標値を得たので、その座標値を基に地図上で該当位置を表示させる
2	地図を表示する(縮尺指定)	適切な表示縮尺で地図を表示させる
3	地図を表示する(表示する地物を指定)	検索結果の地物をハイライト表示させたり、表示する地図に含まれる地物を指定する
4	地図を移動させる	地図を移動させる(スクロールバー、ドラッグ&ドロップ、8方向ボタン等で実現)
5	地図の凡例を表示する	表示されている地図に含まれる地物の凡例を表示する

(2) 地名辞典に関するインタフェース

ここでいう「地名辞典に関するインタフェース」とは、建設情報を共有する場面を想定したとき、地名辞典に関するサービスとして以下の機能を実現するために必要なインタフェースである。

表 2-4 地名辞典に関するインタフェース

No.	機能	具体例
1	地名(地理識別子)を基に、地名に該当する座標を検索(地理識別子と座標の変換)	「赤坂」「東京タワー」「日本武道館」等の地名(地理識別子)にあたるキーワードを基にし、地名に該当する座標を検索する
2	座標を基に、付近の地名(地理識別子)を検索(座標と地理識別子の変換)	GIS 等で取得した座標や座標範囲の値から、座標値の近傍や座標範囲内に含まれる地名(地理識別子)を検索する
3	地名(地理識別子)を基に、地名に関連する別の地名を検索	「千代田区」という地名をキーワードとし、千代田区の下位住所の検索(「神田」や「九段南」など)

通常は、上記の処理を行った後、「地図上で位置を表示する」等の処理を行うが、地名辞典が提供するサービスではなく、地名辞典が実現すべきインタフェースには該当しない為、その後の処理に関してはここでは定義しない

(3) メタデータ検索に関するインタフェース

ここでいう「メタデータ検索に関するインタフェース」とは、建設情報を共有する場面を想定したとき、メタデータ検索に関するサービスとして以下の機能を実現するために必要なインタフェースである。

メタデータ検索とは、特定のデータベースや特定の情報を検索するということではなく、様々な情報がメタデータとして整備され、クリアリングハウスのような仕組みにより、横断的にメタデータの検索を実現する仕組みのことを指す。

表 2-5 メタデータ検索に関するインタフェース

No.	機能	具体例
1	建設情報の検索(キーワード)	「スーパー堤防」「舗装」「環境」等のキーワードを利用し、キーワードに関連する建設情報(電子納品成果)を検索する
2	建設情報の検索(地理的範囲)	上記に加え、地図のインタフェース等を組み合わせ、地理的範囲(座標範囲)も考慮した検索を行う
3	検索結果の表示(候補一覧表示)	キーワードや地理的範囲に該当する建設情報の候補一覧を表示する
4	検索結果の表示(メタデータの表示)	検索結果の候補一覧を1件ずつ選択することで、それぞれのメタデータを確認できる。

2-4. 標準インターフェースの定義方法

標準インターフェースの定義方法を以下に示す。

- (1) 建設情報の連携を実現するうえで必要とされる機能を抽出する。
- (2) 同種の機能を取りまとめ、関数を整理する。
- (3) ISO で提示されているインターフェースについては、それを採用する。
- (4) ISO で提示されているインターフェースだけでは不十分と判断したものについては、独自のインターフェースを定義する。
- (5) 既存の標準インターフェースで、該当する機能が規定されているかどうかを調査し、既存の標準インターフェースで取り扱っている機能は、それを引用する。

【解説】

上記の定義方法を定義するにあたり、基本的な考え方としては、以下の考え方に基づき、既存の標準インターフェースや ISO 等の標準規格を最大限に利用するものとする。

- ISO あるいは JIS による規格を「引用すべき規格」と位置づけ、インターフェースとして定められている既存の標準規格が存在するものに関しては、それを採用する
- ISO 等の標準化機関が策定する標準規格には、「正式な版」と「検討中の版」が存在する場合がある。検討中の版に関しては、今後細部の仕様が変更になる可能性もあるため、引用する際は可能な限り正式な版を採用することとする
- 同様に、ISO や JIS に次ぐ標準化機関として、OGC があるが、ISO ほどの強制力を持たない「業界主導の実装仕様」であることから、該当するインターフェースの規格が ISO で存在しない場合に、OGC の実装仕様を採用することとする

2-5. 関数仕様の考え方

- (1) 標準インタフェースの関数仕様は、本ガイドラインにおいて定める。アプリケーションプログラムの開発・提供を行う者で、本ガイドラインに定めた機能に該当するサービスを開発・提供する場合は、本ガイドラインで定めた関数仕様に従わなければならない。
- (2) 本ガイドラインで定める関数仕様に該当しない関数を用いる必要が生じた場合は、本ガイドラインで定める取り決めに従い、アプリケーションプログラムの開発・提供者が関数仕様を作成する。

【解説】

アプリケーションプログラムの開発・提供者は、開発・提供しようとするアプリケーションプログラムのサービスが、本ガイドラインで定めた関数仕様に該当する場合には、本ガイドラインで定めた関数仕様に従いアプリケーションプログラムの開発・提供を行わなければならない。

開発・提供しようとするアプリケーションプログラムのサービスが、本ガイドラインで定める関数仕様に該当しない場合には、アプリケーションプログラムの開発・提供者は、本ガイドライン第5章・第6章で定める取り決めに従い、関数仕様と実装仕様を作成しなければならない。また作成した関数仕様と実装仕様は、他のアプリケーションプログラムからの利用も可能とする必要があるため、本ガイドライン第8章により公開しなければならない。

2-6. 関数仕様の実装方式

標準インタフェース関数仕様の実装方式として、以下の2種類を定める。

方式1) HTTP による POST/GET を利用した実装

方式2) HTTP による SOAP と WSDL を利用した実装

【解説】

(1) HTTP による POST/GET を利用した実装方式について

標準インタフェース関数仕様の実装方式は、接続方式を考慮する必要がある。アプリケーションプログラム間での通信プロトコルは、インターネット/イントラネット環境に一般的に利用されている HTTP を採用し、接続方式としてはさらに HTTP による POST/GET の2種類の接続方式を採用する。したがって、それぞれの接続方式に対応する実装方式を定める。HTTP のバージョンは 1.0 または 1.1 とするが、推奨は 1.1 とする。

(2) HTTP による SOAP と WSDL を利用した実装方式について

アプリケーションプログラムにおいて種々の機能を実現するためには、(1) の HTTP による POST/GET を利用した接続方式だけでは不十分な場合がある。その場合は、HTTP による SOAP と WSDL を利用した実装方式を採用する。

たとえば、地理情報に関するインタフェースは、ISO19128 を採用することで、方式1の接続方式となるが、地名辞典に関するインタフェースやメタデータ検索に関するインタフェースに関しては、方式2の SOAP と WSDL を利用した接続方式を採用している。

方式2の実装方式の策定にあたっては、OGC における実装仕様を参考にしつつ、OGC が仕様策定時に参照した上位の標準技術 (W3C における Web サービス等) を活用している。

2-7. 標準インターフェースを利用したアプリケーションプログラムの運用形態

標準インターフェースによるアプリケーションプログラムは、次に示す運用形態を想定する。

- (1) 国(各省庁内)や各地方公共団体内など、単一の機関内における運用
- (2) 国の研究機関など省庁をまたいだ機関も含めた運用
- (3) 大学、NPO、民間企業など、一般利用も含めた運用

【解説】

標準インターフェースに準拠したアプリケーションプログラムは、「建設情報」を取り扱うあらゆる機関での運用を想定する。

建設情報を取り扱う機関の代表的なものは国や地方公共団体であり、まずは単一の機関における業務アプリケーションプログラムでの運用を想定する。将来的には、国、地方公共団体、教育・研究機関、NPO、民間企業等の一般の機関にまでその運用範囲が広がる。

このような運用形態を踏まえ、本ガイドラインでは以下の内容を定める。

- インタフェースの公開、管理に関する仕様・取り決め
- インタフェースの接続に関する仕様・取り決め

2-8. インタフェース公開・管理の考え方

(1) インタフェースの公開

- アプリケーションプログラムは、本ガイドラインで定めた関数仕様、実装仕様に基づくインタフェースを公開する。公開したインタフェースを介して、複数の異なるアプリケーションプログラムに対してサービスを提供する。
- 逆にアプリケーションプログラムは、公開されたインタフェースを介して、複数の異なるアプリケーションプログラムに対して、サービスの提供を依頼する。

(2) インタフェースの管理

- アプリケーションプログラムを管理するため、アプリケーションプログラムの開発者は、公開するインタフェースの関数仕様、実装仕様、ならびにアプリケーションプログラムが提供するサービスの内容を文書として作成し、アプリケーションプログラムの管理者に提出する。

【解説】

(1) インタフェースの公開

製品として提供・販売されている GIS 等アプリケーションプログラムのなかには、インタフェース仕様を公開しているものがあり、アプリケーションプログラムの開発者は、公開されたインタフェース仕様に従い、その GIS 等アプリケーションプログラムを利用するアプリケーションプログラムの開発を行うことができる。このとき、インタフェース仕様を公開しているアプリケーションプログラムのことを、特に「サービス」と位置づける。

ただし、そこで公開されているインタフェース仕様は、そのアプリケーションプログラムにのみ適用される独自の仕様であり、たとえそのアプリケーションプログラムと同種の機能を提供するアプリケーションプログラムがあったとしても、機関 A が開発したアプリケーションプログラムが、機関 B のアプリケーションプログラムを利用することは不可能であった。この結果、同様のサービスを提供するアプリケーションプログラムであっても、製品によってインタフェースが異なるためにその都度アプリケーションプログラムを開発するという非効率な状況にあった。このような非効率な状況を避けるため、同様のサービスを提供する場合は、本ガイドラインに基づく共通のインタフェース仕様に基づいた実装を行うものとする。

(2) インタフェースの管理

標準インタフェースが公開され、アプリケーションプログラム間を自由に接続できるようになったとしても、どこにどのようなインタフェースを提供するサービスがあるの

か分からなければ、そのインタフェースを利用することはできない。また、サービスの存在を発見できたとしても、公開されているインタフェースの仕様が不明な場合、インタフェースへの接続はできず、利用することができない。

このような課題を克服するためには、どこにどのようなインタフェースを提供するサービスが存在しているかという情報と、そのインタフェースに接続するためのインタフェース仕様を管理する必要がある。その方法のひとつとして、アプリケーションプログラム管理サーバを設け運用する方法が考えられる。サービスの管理とそれに基づいたインタフェースへの接続の概念図を、図 2-3 に示す。サービスを管理する単位としては、事務所単位、地方整備局単位のほか、将来的には外部機関による管理も考えられる。

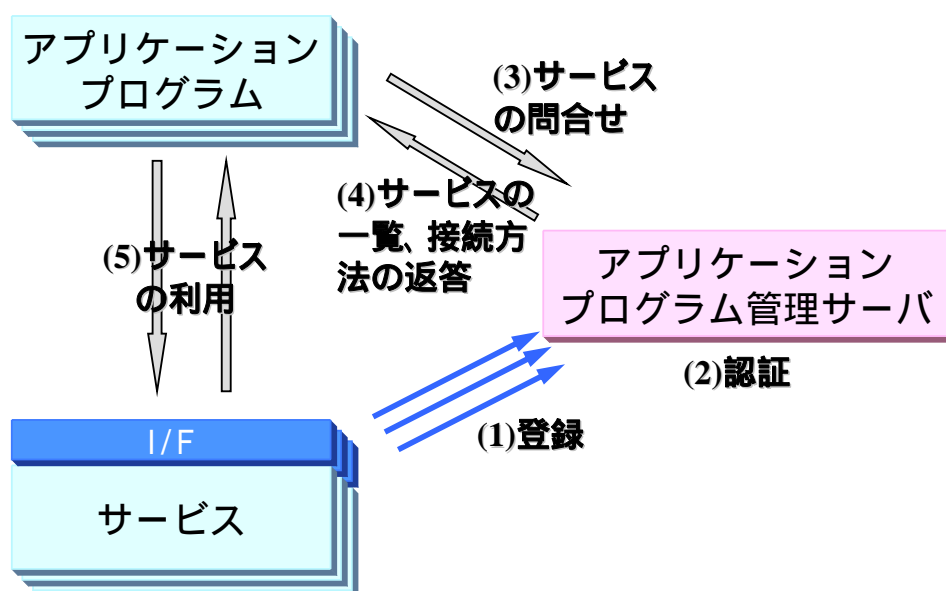


図 2-3 インタフェース管理方法概念図（参考）

2-9. インタフェース接続の考え方

アプリケーションプログラム間での接続は、インタフェースを介して行う。

アプリケーションプログラム間での接続に用いる通信プロトコルは、HTTP とし、以下のうちいずれかの方式とする。

方式1)POST/GET により行う場合

方式2)SOAP により行う場合

アプリケーション間で接続を行う場合には、基本的に以下の手順で行う。

- (1) アプリケーションプログラムとサービスを提供するアプリケーションプログラム間の接続認証
- (2) アプリケーションプログラムによる具体的な命令の送信
- (3) サービスを提供するアプリケーションプログラムによる命令に対する処理結果の送信

【解説】

アプリケーションプログラム間での接続に用いる通信プロトコルは、インターネット/イントラネット環境に一般的に利用されている HTTP を採用する。

HTTP による通信では、HTML 文書の表示等、一般的にクライアントからサーバへの要求に POST/GET メソッドを使う方法が利用されている。また Web サービスにおいても、HTTP などにより XML 形式のデータをやり取りする SOAP の利用が現在の主流である。

そこで本ガイドラインでは、一般的に利用されている POST/GET による接続方式(方式1)と、Web サービスを念頭に、SOAP による接続方式(方式2)の2種類を採用する。

アプリケーションプログラム間では、基本的には次に示す手順に則り接続を行う(図2-4)。

(1) アプリケーションプログラムとサービスを提供するアプリケーションプログラム間の接続認証

- アプリケーションプログラムから、サービスを提供するアプリケーションプログラムに対して、接続の依頼を行う。
- サービスを提供するアプリケーションプログラムは、それに対して接続の認証を行う。
- サービスを提供するアプリケーションプログラムは、認証結果を問合せしたアプリケーションプログラムに返す。

(2) アプリケーションプログラムによる具体的な命令の送信

- アプリケーションプログラムから、サービスを提供するアプリケーション

プログラムに対して、サービスに対する具体的な命令（関数、引数、戻り値の型）を送信する。

(3) サービスを提供するアプリケーションプログラムによる命令に対する処理結果の送信

- サービスを提供するアプリケーションプログラムは、命令に対する処理を行い、その結果について処理を依頼したアプリケーションプログラムに返す。

あるアプリケーションプログラム（サービス利用側）から、処理の命令を受け取ったサービス提供側のアプリケーションプログラムは、その命令が接続認証されたアプリケーションプログラムからのものなのかどうか、どのアプリケーションプログラムからの命令で結果をどこに返すのかを把握しておかなければならない。したがって接続においては、(a)認証に関する情報と、(b)命令の内容や命令に対する処理結果に関する情報により通信を行う。

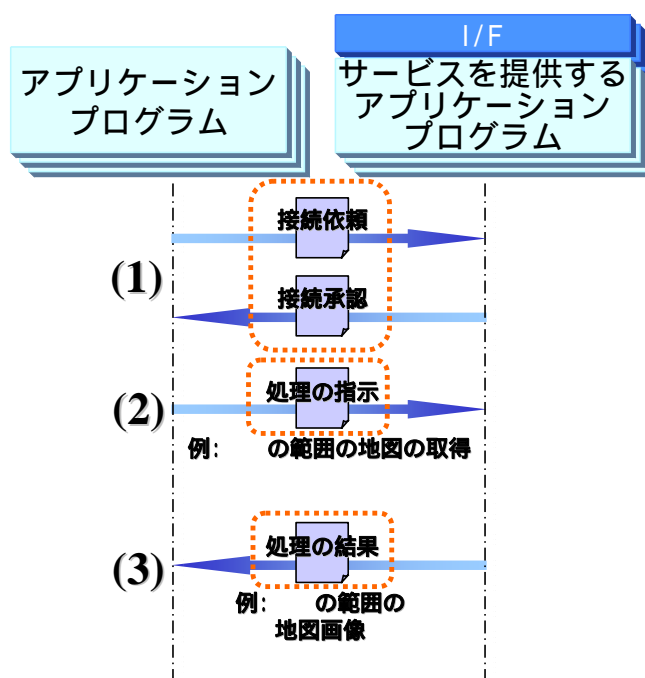


図 2-4 インタフェース接続の手順

SOAP による接続方式（方式 2）においては、SOAP の仕様の解釈の違いから異なるプラットフォームで実装されたインタフェースが接続できないことがあることが知られている。そこで、本ガイドラインでは、相互運用のための指針として、WS-I Basic Profile 1.0 を適用し、実装プラットフォームに依存しないインタフェースを定義する。

3. 標準インタフェース関数仕様

地名辞典サービス、地図サービス、検索サービス等を連携する上で必要となるインタフェースについて、「関数仕様」を定め、解説する。

関数仕様では、以下の項目を定める。

- 関数名
- 戻り値
- 機能
- 関連項目
- 引数
- 説明

それぞれのインタフェースにおいて、必要となる関数は、以下のとおりである。

- A) 提供するサービスの内容を取得するための関数
- B) 情報を検索するための関数
- C) 情報を取得するための関数
- D) 情報を登録するための関数

3-1. 関数の一覧

以下に、本ガイドラインで定義する関数の一覧を示す。

表 3-1 本ガイドラインで定義する関数の一覧

インタフェース種類	種別	必要な機能	機能を実現するための関数
共通	A	サービスメタデータの取得	GetCapabilities
地理情報	B	地図画像の取得	GetMap
	B	(地物の属性情報の取得)	(GetFeatureInfo)
地名辞典 (空間参照系管理)	B	地理識別子による空間参照系の検索	SearchLRS
	C	地理識別子による空間参照系の取得	GetLRS
	D	(地理識別子による空間参照系を新規登録する)	(AddLRS)
	D	(地理識別子による空間参照系を更新する)	(UpdateLRS)
	D	(地理識別子による空間参照系を削除する)	(DeleteLRS)
	B	場所型の検索	SearchLocationType
	C	場所型の取得	GetLocationType
	D	(場所型の新規登録)	(AddLocationType)
	D	(場所型の更新)	(UpdateLocationType)
	D	(場所型の削除)	(DeleteLocationType)
	B	場所型関連の検索	SearchLocationTypeAssociation
	D	(場所型関連の新規登録)	(AddLocationTypeAssociation)
	D	(場所型関連の削除)	(DeleteLocationTypeAssociation)
	地名辞典 (地名辞典管理)	D	(地名辞典の新規登録)
D		(地名辞典の更新)	(UpdateGazetteer)
D		(地名辞典の削除)	(DeleteGazetteer)
B		場所インスタンスの PI 部分の検索	SearchPI
C		場所インスタンスの取得	GetLocationInstance
D		(場所インスタンスの新規登録)	(AddLocationInstance)

インタフェース種類	種別	必要な機能	機能を実現するための関数
	<i>D</i>	(場所インスタンスの更新)	(UpdateLocationInstance)
	<i>D</i>	(場所インスタンスの削除)	(DeleteLocationInstance)
	B	複数場所インスタンス関連の検索	SearchLocationInstanceAssociation
	<i>D</i>	(場所インスタンス関連の新規登録)	(AddLocationInstanceAssociation)
	<i>D</i>	(場所インスタンス関連の削除)	(DeleteLocationInstanceAssociation)
メタデータ検索	C	メタデータで使われる型定義の取得	DescribeRecord
	<i>C</i>	メタデータの属性の定義域の取得	(GetDomain)
	B	検索結果一覧の取得	GetRecords
	C	IDによる検索結果の取得	GetRecordById
	<i>D</i>	メタデータの登録、更新、削除	(Transaction)
	<i>C</i>	メタデータ情報の収集	(Harvest)
<p>【種別凡例】</p> <p>A: 提供するサービスの内容を取得するための関数</p> <p>B: 情報を検索するための関数</p> <p>C: 情報を取得するための関数</p> <p>D: 情報を登録するための関数</p>			

※ 表中のカッコ (斜体) はオプション扱いの関数であり、実装は必須ではない。それ以外の関数に関しては、当該インタフェースを実装する場合は実装を必須とした。本ガイドラインでは、上記に挙げた全ての関数に関して関数仕様を定義したが、実装仕様に関しては必須扱いの関数のみの提示とした。

3-2. 関数の説明

以下より、表 3-1 で示した各関数の概要を示す。

3-2-1. 共通

3-2-1-1. GetCapabilities

機能	サービスメタデータを取得する	
引数	型	名前
	String	version
	説明	
	準拠する標準インタフェースのバージョン番号 該当するバージョンのサービスメタデータのみを取得する 指定なしの場合は、最新バージョン	
	String	service
	サービスタイプ 要求するサービスの識別子 例) 地理情報 “WMS” 地名辞典(空間参照系管理) “SRS” 地名辞典(地名辞典管理) “Gazetteer” メタデータ検索 “CSW”	
String	request	要求名 ”GetCapabilities”でなければならない
String	format	サービスメタデータの出力フォーマット 応答をどのようなフォーマット(MIME Type)で返すかを指定する 省略された場合は、”text/xml”とみなされる
String	updateSequence	サービスメタデータの連番 サービス呼び出し側がこの連番に該当するサービスメタデータをキャッシュしている場合に指定する
戻り値	各 ServiceMetadata	service の内容に応じたサービスメタデータ
関連項目	各 ServiceMetadata 構造体 (※引数”service”において指定したサービスタイプにより、構造体の内容は異なる)	
	サービスタイプ	サービスメタデータ
	WMS	WMSServiceMetadata
	SRS	SRSServiceMetadata
	Gazetteer	GazetteerServiceMetadata
	CSW	CSWServiceMetadata

3-2-2. 地理情報に関するインターフェース

3-2-2-1. GetMap

機能	地図画像を取得する		
引数	型	名前	説明
	String	version	準拠する標準インターフェースのバージョン番号 指定なしの場合は、最新バージョン
	String	request	要求名 ”GetMap”を用いなければならない
	String[]	layers	描画レイヤのリスト
	String[]	styles	描画スタイルのリスト
	String	crs	座標参照系
	Integer[]	bBox	座標範囲
	Integer	width	画像の幅
	Integer	height	画像の高さ
	String	format	画像フォーマット image/jpeg、image/png などの MIME タイプ
	Boolean	transparent	背景透明フラグ true: 透過する false: 透過しない
	String	bgColor	背景色 0x で始まる RGB の 16 進数表現
	String	exceptions	例外報告フォーマット サービスメタデータに記述されているフォーマットの なかから、例外報告フォーマットを指定できる 省略した場合、XML となる
	DateTime	time	時間範囲
Integer	elevation	目的のレイヤの標高 サービスメタデータに有効な標高が記述されている 場合に記述することができる	
Integer	dimension	該当する他の次元値 サービスメタデータに有効な次元が記述されている 場合に記述することができる	
戻り値	地図画像	引数”format”で指定した MIME タイプの画像	
関連項目			

3-2-2-2. GetFeatureInfo

機能	地物の属性情報を表示する		
引数	型	名前	
	String	version	準拠する標準インタフェースのバージョン番号 指定なしの場合は、最新バージョン
	String	request	要求名 ”GetFeatureInfo”を用いなければならない
	GetMap 要求 のパラメータ	—	直前の GetMap 要求で送信したパラメータのうち、version と request を除くすべて
	String[]	query_layers	問合せ対象の一つ以上のレイヤのリスト
	String	info_format	地図情報の出力フォーマット サービスメタデータに記述されている”text/xml”などの MIME タイプ
	Integer	feature_count	情報を取得する対象の地物の個数
	Integer	i	地物の水平方向のピクセル単位の座標値
	Integer	j	地物の垂直方向のピクセル単位の座標値
	String	exceptions	例外報告フォーマット サービスメタデータに記述されているフォーマットのなかから、例外報告フォーマットを指定できる 省略した場合、XML となる
戻り値	地物の情報	info_format で指定した形式で符号化された地物の属性情報	
関連項目			

3-2-3. 地名辞典に関するインタフェース

3-2-3-1. 空間参照系管理

1) SearchLRS

機能	地理識別子による空間参照系を検索する	
引数	型	名前
	String	name
	GeographicExtent	domainOfValidity
	String	theme
	String	overallOwner
	String[]	locationType
	AuthResult	authResult
説明	名称 地理的範囲 分類 管理者 関連する場所型の名称 複数指定された場合は AND 検索を行う 認証結果 ユーザ認証を必要としないサービスの場合は null を指定する	
戻り値	LRS[]	
説明	空間参照系が 1 件もヒットしなかった場合は要素数 0 の配列を返す。	
関連項目	GeographicExtent 構造体、LRS 構造体、AuthResult 構造体	

2) GetLRS

機能	地理識別子による地理識別子による空間参照系を取得する	
引数	型	名前
	String	name
	AuthResult	authResult
説明	名称 認証結果 ユーザ認証を必要としないサービスの場合は null を指定する	
戻り値	LRS	
説明	指定した名称の空間参照系が存在しない場合は null を返す。	
関連項目	LRS 構造体、AuthResult 構造体	

3) AddLRS

機能	地理識別子による空間参照系を新規登録する	
引数	型	名前
	LRS	lrs
	String[]	locationType
	AuthResult	authResult
	SRSExamResult	examResult
	説明	
		登録対象
		関連する場所型の名称 場所型は登録済みのものでなければならない
		認証結果
		審査結果
戻り値	boolean	
説明	登録対象の空間参照系が地理的境界またはポリゴンを持つ場合、その空間参照系は登録済空間参照系の名称である必要がある。 処理に失敗した場合、空間参照系データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	LRS 構造体、AuthResult 構造体、SRSExamResult 構造体	

4) UpdateLRS

機能	地理識別子による空間参照系を更新する	
引数	型	名前
	LRS	lrs
	String[]	locationType
	AuthResult	authResult
	SRSExamResult	examResult
	説明	
		登録対象
		関連する場所型の名称 場所型は登録済みのものでなければならない
		認証結果
		審査結果
戻り値	boolean	
説明	更新対象の空間参照系が地理的境界またはポリゴンを持つ場合、その空間参照系は登録済空間参照系の名称である必要がある。 処理に失敗した場合、空間参照系データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	LRS 構造体、AuthResult 構造体、SRSExamResult 構造体	

5) DeleteLRS

機能	地理識別子による空間参照系を削除する	
引数	型	名前
	String[]	lrs
	AuthResult	authResult
戻り値	boolean	
説明	1件でも空間参照系の削除に失敗した場合は false を返す。 処理に失敗した場合、空間参照系データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	AuthResult 構造体	

6) SearchLocationType

機能	登録されている場所型を検索する		
引数	型	名前	
	String	name	
			名称 場所型の名称を指定する。一部のみの指定も可能である。名称を特定しない場合は null を指定する。
	String	theme	
			分類 一部のみの指定も可能である。分類を特定しない場合は null を指定する
	PiType	identification	
			PIタイプ PIタイプの各要素の値は一部のみの指定が可能
	GeographicExtent	territoryOfUse	
			地理的範囲
String	owner		
		管理者	
String[]	lrs		
		関連する空間参照系の名称 空間参照系を限定しない場合は null を指定する 複数した場合は OR 検索を行う。	
String[]	parentType		
		親の場所型の名称 親関連の条件を指定しない場合は null を指定する	
String[]	childType		
		子の場所型の名称 子関連の条件を指定しない場合は null を指定する	
AuthResult	authResult		
		認証結果 ユーザ認証を必要としないサービスの場合は null を指定する	

戻り値	LocationType[]
説明	<p>lrs に地理識別子による空間参照系の名称を指定することで、特定の空間参照系に関連する場所型を検索することが可能である。</p> <p>また、場所型の Nesting 関連の条件を指定する場合は、parentType または childType に関連する場所型の名称を指定する。</p> <p>地理的範囲が指定された場合は、場所型が持つ地理的範囲と一部でも重なるものを返す。</p> <p>条件に合致する場所型が存在しない場合にはサイズ 0 の配列を返す。</p>
関連項目	PIType 構造体、GeographicExtent 構造体、LocationType 構造体、AuthResult 構造体

7) GetLocationType

機能	登録されている場所型を取得する	
引数	型	名前
	String[]	name
	AuthResult	authResult
		<p>認証結果</p> <p>ユーザ認証を必要としないサービスの場合は null を指定する</p>
戻り値	LocationType[]	
説明	条件に合致する場所型が存在しない場合にはサイズ 0 の配列を返す。	
関連項目	LocationType 構造体、AuthResult 構造体	

8) AddLocationType

機能	場所型を新規登録する	
引数	型	名前
	LocationType	locationType
	AuthResult	authResult
	LocationType-ExamResult	examResult
		<p>登録対象</p> <p>null 指定不可</p> <p>認証結果</p> <p>審査結果</p>
戻り値	boolean	
説明	処理に失敗した場合、空間参照系データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	LocationType 構造体、AuthResult 構造体、LocationTypeExamResult 構造体	

9) UpdateLocationType

機能	登録されている場所型を更新する	
引数	型	名前
	LocationType	locationType
	LocationType-ExamResult	examResult
		説明
		更新対象 登録済みの場所型であること。 null 指定不可
	AuthResult	authResult
		認証結果
		審査結果
戻り値	boolean	
説明	処理に失敗した場合、空間参照系データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	LocationType 構造体、AuthResult 構造体、LocationTypeExamResult 構造体	

10) DeleteLocationType

機能	登録されている場所型を削除する	
引数	型	名前
	String[]	name
	AuthResult	authResult
		説明
		削除対象の場所型の名称 登録済みの場所型であること。
		認証結果
戻り値	boolean	
説明	処理に失敗した場合、空間参照系データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	AuthResult 構造体	

11) SearchLocationTypeAssociation

機能	登録されている場所型関連を検索する。	
引数	型	名前
	String	childLocationType
	String	parentLocationType
	AuthResult	authResult
		説明
		子場所型の名称
		親場所型の名称
		認証結果
		ユーザ認証を必要としないサービスの場合は null を指定する
戻り値	LocationTypeAssociation[]	
説明	子場所型を指定しない場合は親場所型で指定された場所型のすべての子場所型を返す。 親場所型を指定しない場合は子場所型で指定された場所型のすべての親場所型を返す。	
関連項目	LocationTypeAssociation 構造体、AuthResult 構造体	

12) AddLocationTypeAssociation

機能	場所型関連を新規に登録する。	
引数	型	名前
	LocationTypeAssociation[]	association
	AuthResult	authResult
		説明
		登録対象
		認証結果
		ユーザ認証を必要としないサービスの場合は null を指定する
戻り値	boolean	
説明	1件でも登録に失敗した場合には例外 UpdateFailure を発生する。 処理に失敗した場合、空間参照系管理ベースを本関数呼び出し前の状態に戻すこと。	
関連項目	LocationTypeAssociation 構造体、AuthResult 構造体	

13) DeleteLocationTypeAssociation

機能	場所型関連を削除する。	
引数	型	名前
	LocationTypeAssociation	association
	AuthResult	authResult
		説明
		削除対象
		認証結果
		ユーザ認証を必要としないサービスの場合は null を指定する
戻り値	boolean	
説明	削除に失敗した場合は false を返す。 処理に失敗した場合、空間参照系管理サービスを本関数呼び出し前の状態に戻すこと。	
関連項目	LocationTypeAssociation 構造体、AuthResult 構造体	

3-2-3-2. 地名辞典管理

1) AddGazetteer

機能	地名辞典を新規登録する	
引数	型	名前
	Gazetteer	gazetteer
	String	lrs
	説明	登録対象 座標参照系は登録済みのものであること
	説明	参照する空間参照系の名称 登録済みの空間参照系であること
	説明	認証結果
戻り値	boolean	
説明	処理に成功した場合、地名辞典サービスメタデータの連番を更新する。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	Gazetteer 構造体、AuthResult 構造体	

2) UpdateGazetteer

機能	登録されている地名辞典を更新する	
引数	型	名前
	Gazetteer	gazetteer
	AuthResult	authResult
	説明	更新対象 座標参照系は登録済みのものであること
	説明	認証結果
戻り値	boolean	
説明	処理に成功した場合、地名辞典サービスメタデータの連番を更新する。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	Gazetteer 構造体、AuthResult 構造体	

3) DeleteGazetteer

機能	登録されている地名辞典を削除する	
引数	型	名前
	String[]	name
	AuthResult	authResult
	説明	削除対象の地名辞典の名称
	説明	認証結果
戻り値	boolean	
説明	1件でも削除に失敗した場合は false を返す。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	AuthResult 構造体	

4) SearchPI

機能	登録されている場所インスタンスの PI 部分を検索する		
引数	型	名前	説明
	RouteCondition	routeCondition	他サービス呼び出し条件
	String	gazetteer	地名辞典の名称 地名辞典を特定しない場合は null を指定する
	String	locationType	場所型の名称 場所型を特定しない場合は null を指定する
	String[]	geographicIdentifier	地理識別子 地理識別子を特定しない場合は null を指定する
	GeographicExtent	geographicExtent	地理的範囲 地理的範囲を限定しない場合は null を指定する
	TemporalExtent	temporalExtent	時間範囲 時間範囲を限定しない場合は null を指定する
	String	administrator	管理者 管理者を特定しない場合は null を指定する
	String[]	searchWord	検索語句 場所インスタンスの PI の値または代替地理識別子を対象とした全文検索用フリーワード。 語句が複数指定された場合は、地理識別子、または代替地理識別子のいずれかがすべての検索語句を含むものを検索する。 検索語句を設定しない場合は null を指定する。
	AuthResult	authResult	認証結果 ユーザ認証を必要としないサービスの場合は null を指定する
戻り値	PI[]		

<p>説明</p>	<p>指定した条件に合致する PI を検索する。 地理的範囲で地理的境界またはポリゴンを指定した場合は、一部でも地理的範囲が交差する PI を検索する。 条件に合致する PI が存在しない場合はサイズ 0 の配列を返す。 (A) routeCondition.cascadingLevel > 0 の場合 他の地名辞典管理サービスの SearchPI 関数を呼び出す場合は routeCondition.cascadingLevel の値を 1 だけ減じ、かつ routeCondition.routeHistory の末尾に自サーバを追加しなければならない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。 (B) routeCondition.cascadingLevel = 0 の場合 他の地名辞典管理サービスの SearchPI 関数を呼び出すことはできない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。 (C) routeCondition.cascadingLevel < 0 の場合 他の地名辞典管理サービスの SearchPI 関数は routeCondition.routeDefinition の順に呼び出さなければならない。</p>
<p>関連項目</p>	<p>GeographicExtent 構造体、TemporalExtent 構造体、PI 構造体、RouteCondition 構造体、AuthResult 構造体</p>

5) GetLocationInstance

機能	登録されている場所インスタンスを取得する	
引数	型	名前
	RouteCondition	routeCondition
	String	gazetteer
	PI[]	pi
	AuthResult	authResult
戻り値	LocationInstance[]	
説明	<p>指定した PI に合致する場所インスタンスを取得する。 地名辞典を特定して場所インスタンスを取得することもできる。 PI が複数指定された場合は、指定された順に対応する場所インスタンスを返す。PI に対応する場所インスタンスが存在しない場合は、配列中の当該要素に null を格納して返す。 条件に合致する場所インスタンスが存在しない場合はサイズ 0 の配列を返す。</p> <p>(A) routeCondition.cascadingLevel > 0 の場合 他の地名辞典管理サービスの GetLocationInstance 関数を呼び出す場合は routeCondition.cascadingLevel の値を 1 だけ減じ、かつ routeCondition.routeHistory の末尾に自サーバを追加しなければならない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。</p> <p>(B) routeCondition.cascadingLevel = 0 の場合 他の地名辞典管理サービスの GetLocationInstance 関数を呼び出すことはできない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。</p> <p>(C) routeCondition.cascadingLevel < 0 の場合 他の地名辞典管理サービスの GetLocationInstance 関数は routeCondition.routeDefinition の順に呼び出さなければならない。</p>	
関連項目	PI 構造体、LocationInstance 構造体、RouteCondition 構造体、AuthResult 構造体	

6) AddLocationInstance

機能	場所インスタンスを新規登録する		
引数	型	名前	説明
	String	gazetteer	地名辞典の名称
	LocationInstance[]	locationInstance	場所インスタンス 代表点、地理的境界、ポリゴンのいずれかを持つ場合、座標参照系は登録済みのものである必要がある。
	AuthResult	authResult	認証結果
戻り値	Boolean		
説明	1 件でも登録に失敗した場合には例外 UpdateFailure を発生する。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。		
関連項目	LocationInstance 構造体、AuthResult 構造体		

7) UpdateLocationInstance

機能	登録されている場所インスタンスを更新する		
引数	型	名前	説明
	String	gazetteer	地名辞典の名称
	LocationInstance[]	locationInstance	場所インスタンス 代表点、地理的境界、ポリゴンのいずれかを持つ場合、座標参照系は登録済みのものである必要がある。
	AuthResult	authResult	認証結果
戻り値	boolean		
説明	1 件でも更新に失敗した場合には false を発生する。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。		
関連項目	LocationInstance 構造体、AuthResult 構造体		

8) DeleteLocationInstance

機能	登録されている場所インスタンスを削除する		
引数	型	名前	説明
	String	name	地名辞典の名称
	String[]	geographicIdentifier	削除対象場所インスタンスの地理識別子
	AuthResult	authResult	認証結果
戻り値	Boolean		
説明	1 件でも削除に失敗した場合には false を返す。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。		
関連項目	AuthResult 構造体		

9) SearchLocationInstanceAssociation

機能	複数場所インスタンス関連を検索する	
引数	型	名前
	RouteCondition	routeCondition
	String	sourceGazetteer
	String	sourceGeographicIdentifier
	String	targetGazetteer
	String	targetGeographicIdentifier
	String[]	phase
	AuthResult	authResult
戻り値	LocationInstanceAssociation[]	

<p>説明</p>	<p>条件に合致する場所インスタンス関連が存在しない場合はサイズ 0 の配列を返す。</p> <p>(A) routeCondition.cascadingLevel > 0 の場合 他の地名辞典管理サービスの SearchLocationInstanceAssociation 関数を呼び出す場合は routeCondition.cascadingLevel の値を 1 だけ減じ、かつ routeCondition.routeHistory の末尾に自サーバを追加しなければならない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。</p> <p>(B) routeCondition.cascadingLevel = 0 の場合 他の地名辞典管理サービスの SearchLocationInstanceAssociation 関数を呼び出すことはできない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。</p> <p>(C) routeCondition.cascadingLevel < 0 の場合 他の地名辞典管理サービスの SearchLocationInstanceAssociation 関数は routeCondition.routeDefinition の順に呼び出さなければならない。</p> <p>Nesting 関連を検索する場合は source を child に、target を parent に読み替える。</p>
<p>関連項目</p>	<p>LocationInstanceAssociation 構造体、 RouteCondition 構造体、 AuthResult 構造体</p>

10) AddLocationInstanceAssociation

機能	場所インスタンス関連を新規登録する	
引数	型	名前
	LocationInstanceAssociation[]	association
	AuthResult	authResult
説明	登録対象 親および子の場所インスタンスは登録済みのものでなければならない	
戻り値	boolean	
説明	1件でも登録に失敗した場合は false を返す。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	LocationInstanceAssociation 構造体、AuthResult 構造体	

11) DeleteLocationInstanceAssociation

機能	登録されている場所インスタンス関連を削除する	
引数	型	名前
	LocationInstanceAssociation	association
	AuthResult	authResult
説明	削除対象	
戻り値	boolean	
説明	1件でも削除に失敗した場合は false を返す。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	LocationInstanceAssociation 構造体、AuthResult 構造体	

3-2-4. メタデータ検索に関するインタフェース

3-2-4-1. DescribeRecord

機能	レコードの概要(レコードが記述される形式の詳細)を取得する	
引数	型	名前
	String	request
	String	service
	String	version
	String	namespace
	String	typeName
	String	outputFormat
	String	schemaLanguage
戻り値	DescribeRecordResponse	schemaLanguage で指定したスキーマ言語に基づくレコードの構造
関連項目	DescribeRecordResponse 構造体	

3-2-4-2. GetDomain

機能	メタデータの各項目に対する定義域を取得する	
引数	型	名前
	String	request
	String	service
	String	version
	String	parameterName
String	propertyName	プロパティ名
戻り値	GetDomainResponse	指定したパラメータあるいはプロパティのデータ構造を示した XML Schema
関連項目	GetDomainResponse 構造体	

3-2-4-3. GetRecords

機能	検索結果一覧の取得		
引数	型	名前	説明
	String	request	要求名 "GetRecords"でなければならない
	String	service	サービスタイプ "CSW"でなければならない
	String	version	バージョン番号
	String	namespace	レコードが参照する名前空間のリスト
	String	resultType	以下のいずれかより選択: - hits - results - validate
	String	outputFormat	出力ファイルフォーマット。 "text/xml" 等
	String	outputSchema	出力ファイルフォーマット(XML)が参照するスキーマ名
	Integer	startPosition	検索開始位置(カーソル位置)
	Integer	maxRecords	検索結果の件数の上限値
	String	typeName	コレクション ID
	String	elementName / elementSetName	要素名 or 要素体名
	String	constraintLanguage	検索条件式の形式 CQL_TEXT or FILTER(OGC Filter)
	String	constraint	検索条件式(クエリ)
	String[]	sortBy	ソート対象のフィールドと順番のリスト
	Boolean	distributedSearch	検索クエリが既存であるかどうかの論理値 規定値は FALSE
Integer	hopCount	検索クエリの上限值 規定値は"2"	
URL	responseHandler	操作要求の応答状況をモニタするハンドラ	
戻り値	GetRecordsResponse		
関連項目	GetRecordsResponse 構造体		

3-2-4-4. GetRecordById

機能	IDによる検索結果(メタデータ)の取得		
引数	型	名前	説明
	String	version	バージョン番号
	String[]	id	GetRecordsで取得したID(複数指定可能)
戻り値	Metadata	検索結果レコード	
関連項目	Metadata 構造体		

3-2-4-5. Transaction

機能	レコードを追加・更新・削除する		
引数	型	名前	説明
	Transaction	transaction	トランザクションの種類 以下の3つのいずれかを指定する - insert - update - delete
	URI	requested	リクエストID
	Boolean	verboseResponse	レスポンスの種類 TRUE: 通常のレスポンスを要求 FALSE: CSW サーバ固有のレスポンスを要求
戻り値	TransactionResponse	実行したトランザクションの結果を報告する	
関連項目	Transaction 構造体、 TransactionResponse 構造体		

3-2-4-6. Harvest

機能	レコードの更新状況の収集		
引数	型	名前	説明
	String	request	要求名 "Harvest"でなければならない
	String	service	サービスタイプ "CSW"でなければならない
	String	version	準拠する標準インタフェース(CSW)のバージョン番号 指定なしの場合は、最新バージョン
	String[]	namespace	レコードが参照する名前空間のリスト
	URI	source	リソースとなるメタデータを示す URI
	URI	resourceType	リソースとなるメタデータの構造を示す URI 例) リソースとなるメタデータを規定する XML Schema を参照する URI など
	String	resourceFormat	リソースのフォーマット 収集 (Harvest) するリソースのフォーマット (MIME Type) を指定する 省略された場合は、"text/xml"とみなされる
	URL	responseHandler	操作要求の応答状況をモニタするハンドラ
	TemporalExtent	harvestInterval	時間範囲
戻り値	TransactionResponse		
関連項目	TemporalExtent 構造体、 TransactionResponse 構造体		

3-2-5. 構造体

3-2-5-1. 地理情報

1) WMSServiceMetadata

説明	地理情報サービスのメタデータ		
要素	型	名前	説明
	String	name	サービスの名称。 規定では"WMS"でなければならない。
	String	title	サービス提供元を示す名前。 WMS サーバ名などを記述する。
	String	abstract	サービスの概要。 サービス提供元あるいはWMS サーバを簡潔に説明した文章。
	String[]	keywordList	キーワードのリスト。 提供するサービスを端的に表すキーワードのリスト。
	URL	onlineResource	オンラインリソース。 WMS における要求を送信可能な具体的 URL。
	ResponsibleParty	contactInformation	問合せ先情報。 サービス提供元や WMS サーバ開発元など、サービスに関する問合せが可能な連絡先等の情報。 ※ ISO19115 で規定される"CI_ResponsibleParty"クラスを引用しており、住所、電話番号等の情報を含む。
	String	fees	サービスの利用における課金の有無。 ※有料である場合は、必ず示されなければならない。
	Integer	layerLimit	一回の GetMap 要求で指定できるレイヤーの上限数。 ※上限が存在する場合は、必ず示されなければならない。
	Integer	maxWidth	WMS サーバが生成する地図画像サイズ(幅)の上限値。 ※上限が存在する場合は、必ず示されなければならない。

	Integer maxHeight	WMS サーバが生成する地図画像サイズ (高さ)の上限値。 ※上限が存在する場合は、必ず示されな なければならない。
	OperationMet request adata[]	GetCapabilities 、 GetMap 、 GetFeatureInfo それぞれに関する詳細 情報
	String[] exception	例外報告フォーマットに指定可能なフォー マットの情報 例外報告を行う際に指定可能なフォー マット
	LayerMetadat layer a	地図画像生成元となるレイヤーに関する 情報
関連項目	ResponsibleParty 構造体、OperationMetadata 構造体、LayerMetadata 構造体	

2) LayerMetadata

説明	地図画像生成元となるレイヤーに関する情報		
要素	型	名前	説明
	String	name	レイヤーの名称
	String	title	レイヤーのタイトル ※GetMap 要求では、Name ではなく Title によりレイヤー名を指定するため、Title は必須である。
	String	abstract	レイヤーの概要 レイヤーに含まれる地物の情報や、主題の名称
	String[]	keywordList	キーワードリスト レイヤーを端的にあらわすキーワードのリスト
	String	crs	レイヤーに適用される座標参照系 レイヤーCRS の座標参照系識別子
	Double[]	EX_Geographic BoundingBox	地理境界ボックス レイヤー全体の東西南北の境界緯度経度 ※絶対緯度経度による記述
	GeographicBoundingBox[]	boundingBox	境界ボックス レイヤー全体の境界座標 ※座標参照系を含み、座標参照系ごとの記述が可能
		dimension	次元 レイヤーが属する次元。※さらに子要素を持つクラスである
	InfoURL	attribution	情報提供元の属性情報
	InfoURL	authorityURL	作者のリソース情報
	String	identifier	レイヤーを識別する ID(識別子)
	InfoURL	metadataURL	メタデータのリソース情報
	InfoURL	dataURL	実データのリソース情報
	InfoURL	featureListURL	地物リストのリソース情報
	InfoURL	style	描画スタイルのリソース情報
	Integer	minScaleDenominator	最小スケールの分母 最小表示スケールが存在する場合、その分母の数値
Integer	maxScaleDenominator	最大スケールの分母 最大表示スケールが存在する場合、その分母の数値	

	Layer layer	子レイヤの情報 ※レイヤーは、子レイヤーを持つことができ、階層構造を構成できる
関連項目	GeographicBoundingBox 構造体、InfoURL 構造体、Layer 構造体	

3) InfoURL

説明	画像リソースの URL 情報		
要素	型	名前	説明
	String	name	名称
	String	title	リソースのタイトル
	String	abstract	リソースの概要
	Integer	width	画像の幅(pixel) ※リソースが画像の場合
	Integer	height	画像の高さ(pixel) ※リソースが画像の場合
	String	format	リソースの MIME タイプ
	URL	onlineResource	リソースの URL
	InfoURL	url	リソースの URL 情報
関連項目			

3-2-5-2. 地名辞典(空間参照系管理)

1) SRSServiceMetadata

用途	空間参照系管理サービスのメタデータ	
要素	型	名前
	String	version
	boolean	canUpdateLRS
	boolean	canUpdateCRS
	boolean	canUpdateConversionInfo
	boolean	canUpdateLocationType
	boolean	needsExam
	String	updateSequence
説明	更新連番は任意の文字列であるが、キャッシュ機能を実現するために、サービスは更新連番の値の大小を判断できることが必要である。	
関連項目		

2) LRS

用途	地理識別子による空間参照系	
要素	型	名前
	String	name
	GeographicExtent	domainOfValidity
	String	theme
	String	overallOwner
説明		
関連項目	GeographicExtent 構造体	

3) SRSExamResult

用途	空間参照系審査結果		
要素	型	名前	説明
	String	name	空間参照系名称
	String	examNo	審査結果番号
説明			
関連項目			

4) PIType

用途	PIタイプ		
要素	型	名前	説明
	String	schemaUrl	PI スキーマ文書の URL
	String	encodingRuleUrl	PI 符号化規則文書の URL
	String	generationRuleUrl	PI 符号化規則作成規則文書の URL
説明			
関連項目			

5) LocationType

用途	場所型		
要素	型	名前	説明
	String	name	名称
	String	theme	分類
	PIType	identification	PIタイプ
	String	definition	定義
	GeographicExtent	territoryOfUse	地理的範囲
	String	owner	管理者
	String	name	名称
	String	theme	分類
説明			
関連項目	PIType 構造体、GeographicExtent 構造体		

6) LocationTypeAssociation

用途	場所型関連		
要素	型	名前	説明
	String	childLocationType	子場所型の名称 null 指定不可
	String	parentLocationType	親場所型の名称 null 指定不可
説明			
関連項目			

7) LocationTypeExamResult

用途	場所型審査結果		
要素	型	名前	説明
	String	name	場所型名称
	String	examNo	審査結果番号
説明			
関連項目			

3-2-5-3. 地名辞典(地名辞典管理)

1) GazetteerServiceMetadata

用途	地名辞典サービスメタデータ		
要素	型	名前	説明
	String	version	地名辞典サービスのバージョン 本仕様によるサービスメタデータでは 1.0 を指定する。
	Gazetteer[]	gazetteer	地名辞典情報
	GazetteerStatistics[]	gazetteerStatistics	地名辞典統計情報
	boolean	canUpdateLocationInstance	場所インスタンスの登録可否
	boolean	canUpdateLocationInstanceAssociation	場所インスタンス関連の登録可否
	String	updateSequence	連番
説明			
関連項目	Gazetteer 構造体、GazetteerStatistics 構造体		

2) Gazetteer

用途	地名辞典		
要素	型	名前	説明
	String	name	名称
	String[]	alias	別名
	String	scope	スコープ
	GeographicExtent	territoryOfUse	地理的範囲
	String	custodian	管理者
	String	crs	座標参照系
	TemporalExtent	date	時間範囲
説明			
関連項目	GeographicExtent 構造体、TemporalExtent 構造体		

3) GazetteerStatistics

用途	地名辞典統計情報		
要素	型	名前	説明
	String	gazetteer	地名辞典名称
	String	locationType	場所型名称
	Integer	count	総データ件数
説明			
関連項目			

4) PI

用途	PI		
要素	型	名前	説明
	String	value	地理識別子 null は指定不可
	String	lrs	地理識別子による空間参照系の名称 null は指定不可
説明	現実世界のある場所を一意に識別するための識別子。		
関連項目			

5) LocationInstance

用途	場所インスタンス		
要素	型	名前	説明
	PI	PI	PI 地理識別子と空間参照系の組 null は指定不可
	String	locationType	場所型名称 場所型の実体は空間参照系管理サービス が保持する null は指定不可
	String[]	alternative- GeographicIdentifi er	代替地理識別子 地理識別子の別名 代替地理識別子がない場合は null を指定 する
	Geographic Extent	geographicExtent	地理的範囲
	Position	position	位置 代表点の座標
	String	administrator	管理者 場所インスタンスの特質を定義する責任があ る組織の名称 null は指定不可
	TemporalE xtent	temporalExtent	時間範囲 時間範囲を限定しない場合は null を指定す る
	String	gazetteer	地名辞典の名称
説明			
関連項目	PI 構造体、GeographicExtent 構造体、Position 構造体、TemporalExtent 構造体		

6) LocationInstanceAssociation

用途	場所インスタンス関連		
要素	型	名前	説明
	String	sourceGazetteer	関連元場所インスタンスの地名辞典の名称
	String	sourceGeographicIdentifier	関連元場所インスタンスの地理識別子
	String	targetGazetteer	関連先場所インスタンスの地名辞典の名称
	String	targetGeographicIdentifier	関連先場所インスタンスの地理識別子
	String	phase	位相の名称
説明	phase には“Nesting”または“Equivalent”を指定することができる。		
関連項目			

7) RouteCondition

用途	地名辞典管理サービス呼び出し経路条件		
要素	型	名前	説明
	Integer	cascadingLevel	他サービス呼び出し可能レベル 0 以下: 他サービス呼び出さない 1 以上: 他サービスを最大 N 段階呼び出すことができる
	Route[]	routeHistory	他サービス呼び出し経路履歴
	Route[]	routeDefinition	他サービス呼び出し経路定義
説明			
関連項目			

8) Route

用途	地名辞典管理サービスの呼び出し経路		
要素	型	名前	説明
	String	serviceName	地名辞典管理サービスの名称
説明			
関連項目			

3-2-5-4. メタデータ検索

1) CSWServiceMetadata

用途	カタログサービスのサービスメタデータ		
要素	型	名前	説明
	String	name	サービスの名称。 規定では"CWS"でなければならない。
	String	title	サービス提供元を示す名前。 CWS サーバ名などを記述する。
	String	abstract	サービスの概要。 サービス提供元あるいは CWS サーバを簡潔に説明した文章。
	String[]	keywordList	キーワードのリスト。 提供するサービスを端的に表すキーワードのリスト。
	URL	onlineResource	オンラインリソース。 CWS における要求を送信可能な具体的な URL。
	Responsible Party	contactInformation	問合せ先情報。 サービス提供元や CWS サーバ開発元など、サービスに関する問合せが可能な連絡先等の情報。 ※ ISO19115 で規定される"CI_ResponsibleParty"クラスを引用しており、住所、電話番号等の情報を含む。
	String	fees	サービスの利用における課金の有無。 ※有料である場合は、必ず示されなければならない。
	OperationMetadata[]	operationsMetadata	CWS サーバが提供可能な操作についてのメタデータ
Filter_Capabilities	filter_Capabilities	CWS サーバが処理可能な Transaction クエリに関するメタデータ ※OGC Filter1.0 に基づく	
説明	カタログサービスが提供可能なサービスのメタデータを返す		
関連項目	ResponsibleParty 構造体、OperationsMetadata 構造体、Filter_Capabilities 構造体		

2) DescribeRecordResponse

用途	検索対象メタデータの構造		
要素	型	名前	説明
	URI	targetNamespace	スキーマの規定名前空間
	String	parentSchema	親スキーマのファイル名称
	String	schemaLanguage	スキーマの言語
説明	検索対象メタデータの構造を明らかにするために必要な XML Schema の情報を返す		
関連項目			

3) Filter_Capabilities

用途	CWS サーバが処理可能な Transaction クエリに関するメタデータ		
要素	型	名前	説明
	Logical operators	and	CSW サーバが処理可能な論理演算
		or	
		not	
	Comparison operators	propertyIsEqualTo	CSW サーバが処理可能な比較演算
		propertyIsNotEqualTo	
		propertyIsLessThan	
		propertyIsGreaterThan	
		propertyIsLessThanOrEqualTo	
		propertyIsGreaterThanOrEqualTo	
Spatial operators	BBox	CSW サーバが処理可能な空間演算	
説明	CWS サーバが処理可能な Transaction クエリに関するメタデータの情報を返す		
関連項目			

4) GetDomainResponse

用途	メタデータ属性値の項目 (Property) と定義域 (Values)		
要素	型	名前	説明
	String	parameterName	項目名
	String[]	value	項目名に対応する値のリスト
説明	検索対象メタデータの各属性値の情報を返す		
関連項目			

5) GetRecordsResponse

用途	検索結果		
要素	型	名前	説明
	String	requestId	リクエスト ID
	String	status	検索結果ステータス
	String	resultSetId	検索結果セット ID
	String	elementSet	要素セット名称
	String	recordSchema	レコードの書式 (XML Schema 等のスキーマを特定する名称)
	Integer	numberOfRecordsMatched	検索ヒット総数
	Integer	numberOfRecordsReturned	検索結果に含まれる総数
	Integer	nextRecord	次の検索結果リストの始まりのレコード位置
Metadata	resultRecord	検索結果レコード	
説明	検索結果メタデータの概要情報を返す		
関連項目	Metadata 構造体		

6) Metadata

用途	検索結果メタデータの詳細		
要素	型	名前	説明
		any	検索結果メタデータレコードにより異なる
説明	✓ 検索結果レコードである Metadata 構造体は、GetRecordResponse 構造体におけるパラメータ recordSchema で示されるスキーマによってその構造が決定される。		
関連項目			

7) Transaction

(i) Transaction.Insert

用途	トランザクションクエリ(Insert)		
要素	型	名前	説明
	String	transactionType	“update”
	String	requestId	リクエスト ID
	Record	record	処理対象レコード
	Boolean	verboseResponse	レスポンスの種類 TRUE: 通常のレスポンスを要求 FALSE: CSW サーバ固有のレスポンスを 要求
String	handle	操作要求の応答状況	
説明	Insert 処理のリクエストクエリを記述		
関連項目	Record 構造体		

Record 構造体は、処理対象レコードの構造によりその都度異なる

(ii) Transaction.Update

用途	トランザクションクエリ(Update)		
要素	型	名前	説明
	String	transactionType	"update"
	String	requestId	リクエストID
	Record	record	処理対象レコード
	Boolean	verboseResponse	レスポンスの種類 TRUE: 通常のレスポンスを 要求 FALSE: CSW サーバ固有 のレスポンスを要求
	String	recordProperty	レコードプロパティ ※Record がリクエストされな い場合
	String	typeNameNames	処理対象レコードの種類 例) "Service"、 "Dataset"、 Dataset Collection"、 "Application" 等
	Codelist	constraintLanguage	規定値"Filter"
	String	constraint	OGC Filter1.0.0に基づく符 号化規則によるクエリ式
	String	handle	操作要求の応答状況
説明	Update 処理のリクエストクエリを記述		
関連項目	Record 構造体		

Record 構造体は、処理対象レコードの構造によりその都度異なる

(iii) Transaction.Delete

用途	トランザクションクエリ(Delete)		
要素	型	名前	説明
	String	transactionType	"delete"
	String	requestId	リクエスト ID
	Record	record	処理対象レコード
	Boolean	verboseResponse	レスポンスの種類 TRUE: 通常のレスポンスを要求 FALSE: CSW サーバ固有のレスポンスを要求
	String	recordProperty	レコードプロパティ ※Record がリクエストされない場合
	String	typeNameNames	処理対象レコードの種類 例) "Service"、"Dataset"、Dataset Collection"、"Application" 等
	Codelist	constraintLanguage	規定値"Filter"
	String	constraint	OGC Filter1.0.0 に基づく符号化規則によるクエリ式
String	handle	操作要求の応答状況	
説明	Delete 処理のリクエストクエリを記述		
関連項目	Record 構造体		

Record 構造体は、処理対象レコードの構造によりその都度異なる

8) TransactionResponse

用途	トランザクションの結果		
要素	型	名前	説明
	String	requestId	リクエスト ID
	Integer	totalInserted	Insert 成功数(※Insert の場合)
	Integer	totalUpdated	Update 成功数(※Update の場合)
	Integer	totalDeleted	Delete 成功数(※Delete の場合)
	String	handleRef	操作 ID
	Record	briefRecord	Insert/Update/Delete した内容の詳細
説明	トランザクションの結果を報告する		
関連項目	Record 構造体(※Record 構造体は、Insert/Update/Delete したレコードの構造をそのまま報告する)		

3-2-5-5. 共通仕様

1) AuthResult

用途	ユーザ認証結果		
要素	型	名前	説明
	String	token	認証トークン
説明	サービスが発行する、ユーザ認証の結果を一意に特定する文字列を保持する。この文字列を認証トークンと呼ぶことにする。 関数呼び出しの際にユーザ認証結果を受け取ることで、サービスは利用ユーザを特定でき、ユーザごとに機能の利用を制限することができる。		
関連項目			

2) ContactAddress

説明	住所情報		
要素	型	名前	説明
	String	addressType	住所の型
	String	address	住所
	String	city	市町村
	String	stateOrProvince	州、郡、都道府県
	String	postCode	郵便番号
	String	country	国
関連項目			

3) ContactPersonPrimary

説明	問合せ先名称情報		
要素	型	名前	説明
	String	contactPerson	問合せ先個人名
	String	contactOrganization	問合せ先組織名
関連項目			

4) Datum

用途	原子		
要素	型	名前	説明
	String	datumID	原子の識別子
	String[]	alias	原子の別名
	String	type	原子の型
	String	point	地球に固定するために使用する点
	DateTime	realizationEpoch	原子の実現時期
	Extent	validArea	有効な地域
	String	scope	有効な応用分野
	String	remark	注釈
	PrimeMeridian	primeMeridian	本初子午線情報
Ellipsoid	ellipsoid	楕円体情報	
説明			
関連項目	Extent 構造体、PrimeMeridian 構造体、Ellipsoid 構造体		

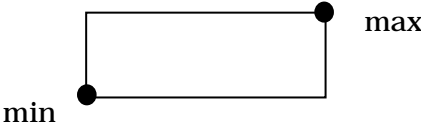
5) Ellipsoid

用途	楕円体		
要素	型	名前	説明
	String	ellipsoidID	原子に対する楕円体の識別子
	String[]	alias	別名
	Double	semiMajorAxis	楕円体の長半径の長さ
	boolean	ellipsoidShape	楕円体の計上
	InverseFlattening	inverseFlattening	楕円体の逆扁平率
	String	remark	注釈
説明			
関連項目	InverseFlattening 構造体		

6) Extent

用途	範囲情報		
要素	型	名前	説明
	String	description	記述
	GeographicExtent[]	geographicElement	地理的範囲
	TemporalExtent[]	temporalElement	時間範囲
	VerticalExtent[]	verticalElement	鉛直範囲
説明			
関連項目	GeographicExtent 構造体、TemporalExtent 構造体、VerticalExtent 構造体		

7) GeographicBoundingBox

用途	地理的境界		
要素	型	名前	説明
	double[]	min	最小座標
	double[]	max	最大座標
	String	crs	座標参照系
説明	<p>矩形で領域を定義する。min と max は下図で示す位置の座標を指定する。</p>  <p>座標配列には座標参照系で定義された順番で値を格納する。 各座標は以下の条件を満たす必要がある。 $\min[i] \leq \max[i]$ (i: 配列のインデックス)</p>		
関連項目			

8) GeographicExtent

用途	地理的範囲		
要素	型	名前	説明
	Polygon	boundingPolygon	ポリゴン
	GeographicBoundingBox	boundingBox	矩形
	String	description	記述
説明	boundingPolygon、 boundingBox、 description のいずれか1つのみ null 以外の値を指定する。		
関連項目	Polygon 構造体、GeographicBoundingBox 構造体		

9) InverseFlattening

用途	楕円体扁平率		
要素	型	名前	説明
	Double	inverseFlattening	逆扁平率
	boolean	isInfinite	
説明			
関連項目			

10) OperationMetadata

説明	サービスが提供可能な操作 (Operation) の詳細情報		
要素	型	名前	説明
	String	operation	提供可能な操作の名称
	String[]	format	応答のファイル形式 (MIME Type)
	URL[]	onlinerResource	Get メソッド/POST メソッドの要求先 URL (※ POST メソッドの URL は、POST メソッドを実装している場合のみ)
	String	Parameter.name	操作が持つパラメータの名称
	String	Parameter.value	操作が持つパラメータの名称に対応する値
	String	extendedCapabilities	本体持つ操作に対し、実装者が拡張した内容
関連項目			

11) Polygon

用途	ポリゴン		
要素	型	名前	説明
	Ring	exterior	外周 null を指定してはならない
	Ring[]	interior	内周 穴あきポリゴンを定義する場合に指定する。穴あきポリゴンでない場合は null を指定する。
説明	1 つの外周と複数の内周からなる多角形を定義する。 外周や内周が内部でもつ座標参照系はすべて同一のものでなくてはならない。		
関連項目	Ring 構造体		

12) Position

用途	位置		
要素	型	名前	説明
	double[]	coord	座標
	String	crs	座標参照系
説明	座標配列には空間参照系で定義された順番で値を格納する。		
関連項目			

13) PrimeMeridian

用途	本初子午線		
要素	型	名前	説明
	String	meridianID	本初子午線の識別子
	Double	GreenwichLongitude	本初子午線グリニッジ経度
	String	remark	注釈
説明			
関連項目			

14) ResponsibleParty

説明	問合せ先情報		
要素	型	名前	説明
	ContactPersonPrimary	contactPersonPrimary	問合せ先名称情報
	String	contactPosition	問合せ先の役職
	ContactAddresses	contactAddress	問合せ先住所情報
	String	contactVoiceTelephone	問合せ先電話番号
	String	contactFacsimileTelephone	問合せ先 FAX 番号
	String	contactElectronicMailAddresses	問合せ先電子メールアドレス
関連項目	ContactPersonPrimary 構造体、ContactAddress 構造体		

15) Ring

用途	リング		
要素	型	名前	説明
	Position[]	coords	座標列
説明	交差しない閉図形を定義する。始点と終点は一致する必要がある。座標列は 3 つ以上の点を含まなければならない。		
関連項目	Position 構造体		

16) TemporalExtent

用途	時間範囲		
要素	型	名前	説明
	DateTime	begin	開始時点 終了時点以前のすべての期間を指定する場合は null を指定する
	DateTime	end	終了時点 開始時点以後のすべての期間を指定する場合は null を指定する
説明	begin、end のいずれか1つは null でない値を指定しなければならない。 また、begin、end ともに値を指定する場合は $begin \leq end$ でなければならない。		
関連項目			

17) VerticalExtent

用途	鉛直範囲		
要素	型	名前	説明
	Double	minimumValue	最低値
	Double	maximumValue	最大値
	String	unitOfMeasure	測定単位
	Datum	verticalDatum	鉛直原子
説明			
関連項目	Datum 構造体		

3-2-5-6. 例外

1) InvalidUpdateSequence

用途	サービスメタデータの更新連番の異常
説明	更新連番の大小関係が異常な場合や大小関係を評価できない場合に発生する。 例外の表現方法は処理系ごとに定義する共通仕様に従う。
関連項目	GetCapabilities 関数

2) InvalidArgument

用途	関数の引数に不正な値が指定された場合の例外
説明	null を許可しない引数に null が指定された場合に発生する。 例外の表現方法は処理系ごとに定義する共通仕様に従う。
関連項目	すべての関数

3) InvalidValue

用途	構造体に不正な値が指定された場合の例外
説明	null を許可しない構造体要素に null が指定された場合に発生する。ただし、関数の引数に指定された構造体の場合は InvalidArgument を発生する。 例外の表現方法は処理系ごとに定義する共通仕様に従う。
関連項目	すべての関数

4) UpdateFailure

用途	データベースの追加、更新、削除に失敗した場合の例外
説明	各種更新系の処理で失敗した場合に発生する。 例外の表現方法は処理系ごとに定義する共通仕様に従う。
関連項目	データベースのトランザクション処理を行うすべての関数

4. 標準インタフェース実装仕様

3章で示した関数仕様に対し、アプリケーションプログラムとして実装を行う場合に必要となる各関数の機能の説明、利用条件、リクエストの内容、レスポンスの内容、例外処理、および各構造体実装時の注意事項について示す。

4-1節では、POST/GET 利用のための実装仕様について示す。

4-2節では、SOAP と WSDL 利用のための実装仕様について示す。

4-1. POST/GET 利用のための実装仕様

4-1-1. 地理情報に関するインタフェース

地理情報に関するインタフェースにおける POST/GET 利用のための実装仕様は、ISO19128 Web Map Server Interface に準ずる。POST/GET 利用のためのインタフェースを実装する場合には、ISO19128 Web Map Server Interface のレスポンスの規則、座標系、リクエストパラメータの規則、各関数の記述規則等に従うものとする。

ISO19128 において規定している Web Map Server Interface のプロトコルバージョンは、「1.3.0」である。

4-1-2. 地名辞典に関するインタフェース

地名辞典に関するインタフェースにおける POST/GET 利用の実装仕様は、特に定めがない。

本ガイドラインにおいては、地名辞典に関するインタフェースに関しては、4-2-4 に示す SOAP と WSDL 利用の実装を推奨する。

4-1-3. メタデータ検索に関するインタフェース

メタデータ検索に関するインタフェースにおける POST/GET 利用の実装仕様は、OGC Catalogue Service Specification において規定されている、HTTP を通信プロトコルとした実装仕様である Catalogue Services for the Web (CSW) に準ずる。POST/GET 利用のためのインタフェースを実装する場合には、CSW のレスポンスの規則、リクエストパラメータの規則、各関数の記述規則等に従うものとする。

CSW において規定しているプロトコルバージョンは、「2.0.0」である。

4-2. SOAP と WSDL 利用のための実装仕様

4-2-1. 共通事項

4-2-1-1. トランスポートプロトコル

SOAP1.1 では、通信プロトコルとして、HTTP のほか、SMTP、FTP などの利用が可能であるが、本ガイドラインでは、HTTP での SOAP 通信を実装することとする。

4-2-1-2. 例外処理

無効なリクエストが発行された場合、実装されていないサービスが要求された場合などは、例外処理として、Fault 要素を含む SOAP メッセージを送付する。Fault 要素には子要素として detail 要素を設け、以下のアプリケーション固有のメッセージを付与するものとする。

```
<someFault>
  <message>メッセージ</message>
  <code>コード</code>
</someFault>
（someFault は例外の内容を表す任意の文字列）
```

代表的な例として以下のものを定義する。

コード	メッセージ
999	この関数は未実装です。
001	パラメータが不正です。
002	セッションがタイムアウトしました。

使用するすべてのコードはインタフェース実装ごとに ServiceMetadata 構造体で定義する。

4-2-1-3. 座標参照系

本ガイドラインで取り扱う座標系は、JIS X 7115 の 付属書 2（規定） 「日本における座標参照系の表記」に従い、測地原子と座標系の組み合わせで表記し、表 4-1 のとおりとする。

表 4-1 本ガイドラインで取り扱う座標系

必須	座標参照系	内容
	JGD2000 / (B, L)	日本測地系 2000 / 緯度経度
	JGD2000 / (B, L, h)	日本測地系 2000 / 緯度経度および高さ ^{※1)}
○	JGD2000, TP / (B, L), H	日本測地系 2000 / 緯度経度および高さ ^{※2)}
	JGD2000 / (X, Y, Z)	日本測地系 2000 / 3次元座標
	WGS84 / (B, L)	世界測地系 / 緯度経度
	WGS84 / (B, L, h)	世界測地系 / 緯度経度および高さ ^{※1)}
	WGS84, TP / (B, L), H	世界測地系 / 緯度経度および高さ ^{※2)}
	WGS84 / (X, Y, Z)	世界測地系 / 3次元座標
	JGD2000 / [Zone No.] ^{※3)} (X, Y)	日本測地系 2000 / 平成 14 年国土交通省告示第 9 号に定める平面直角座標系
	JGD2000 / [Zone No.] ^{※4)} (E, N)	日本測地系 2000 / UTM 座標系
	JGD2000, TP / [Zone No.] ^{※3)} (X, Y), H	日本測地系 2000 / 平成 14 年国土交通省告示第 9 号に定める平面直角座標系(高さを含む) ^{※2)}
	JGD2000, TP / [Zone No.] ^{※4)} (E, N), H	日本測地系 2000 / UTM 座標系(高さを含む) ^{※2)}
	WGS84 / [Zone No.] ^{※3)} (X, Y)	世界測地系 / 平成 14 年国土交通省告示第 9 号に定める平面直角座標系
	WGS84 / [Zone No.] ^{※4)} (E, N)	世界測地系 / UTM 座標系
	WGS84, TP / [Zone No.] ^{※3)} (X, Y), H	世界測地系 / 平成 14 年国土交通省告示第 9 号に定める平面直角座標系(高さを含む) ^{※2)}
	WGS84, TP / [Zone No.] ^{※4)} (E, N), H	世界測地系 / UTM 座標系(高さを含む) ^{※2)}

1：楕円体高

2：東京湾平均海面からの高さ

3：平成 14 年国土交通省告示第 9 号に定める平面直角座標系で用いられる 1～19 までのゾーン No.のいずれか

4：UTM 座標系で用いられる 1～60 までのゾーン No.のいずれか

4-2-1-4. データタイプ

エンコーディングは、XML Schema Part2 に準拠する。なお本ガイドライン中で取り扱われている変数のデータ型は、同規格に準拠したデータ型（XML Schema DataType）とする（表 4-2）。

表 4-2 データ型

ガイドライン表記データ型	XML Schema DataType
整数	long
実数	double
文字	string
日時	dateTime
論理	boolean
バイト	base64Binary

4-2-1-5. 文字エンコーディング

SOAP メッセージの文字エンコーディングは Basic Profile 1.0 で規定されている以下のものを採用する。

- UTF-8
- UTF-16

4-2-2. 各サービスに共通するインタフェース

4-2-2-1. GetCapabilities

1) 機能説明

サービス提供者（以下、プロバイダ）が提供するサービスの内容（サービスメタデータ）を取得する。主な用途は、以下のとおり。

- サービス利用者（以下、リクエスタ）が取扱可能な標準インタフェースのバージョンと、サービス提供者（以下、プロバイダ）が提供可能な標準インタフェースのバージョンの互換性確認を行う。
- プロバイダが提供するサービスに関しての利用条件を把握する。
- なおサービスメタデータは、プロバイダが提供するサービスの内容についてを示したものである。サービスメタデータの内容に関わるサービス更新がされた場合（地図サービスにおけるレイヤ追加など）は、メタデータが更新され、追隨して連番が加算される。

2) 利用条件

リクエスタは、サービス接続直後にバージョンを取得し互換性を確認する。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
String	version	指定しない場合は Null を入力する	リクエスタが利用する標準インタフェースのバージョン番号を指定する
String	service	必須	要求するサービスの識別子を指定する
String	request	必須	要求名"GetCapabilities"を指定する
String	format	指定しない場合は Null を入力する	サービスメタデータの出カフォーマットを指定する
String	updateSequence	指定しない場合は Null を入力する	リクエスタがすでにキャッシュしているサービスメタデータの連番を指定する

(i) version

リクエスタが使用可能な共通インタフェースのバージョン番号を指定する。バージョン番号を指定しない場合は、プロバイダが提供可能なバージョンのうち、最新のバージョンを指定したことと同様の動作となる。

リクエスタが指定したバージョン番号とプロバイダが返すバージョン番号の関係は、以下のとおり。

リクエストが指定するバージョン番号	プロバイダが返すバージョン番号
指定なし	提供可能なバージョンのうち最新のバージョン番号を返す
指定あり (プロバイダが提供可能なバージョン以下のバージョンを指定した場合)	指定されたバージョン以下での最新のバージョン番号を返す ・指定したバージョンが存在する場合は、指定したバージョン番号を返す ・指定したバージョン以下のバージョンが存在しない場合は、提供可能なバージョンのうち最も古いバージョン番号を返す
指定あり (プロバイダが提供可能なバージョンより大きいバージョンを指定した場合)	提供可能なバージョンのうち最新のバージョン番号を返す

【書式】

共通インタフェースバージョン番号は、小数点一桁までの数字で定義される。

(ii) service

要求するサービスの識別子を指定する。

【書式】

service は、以下のいずれかの値によって定義される。

- 地理情報 “WMS”
- 地名辞典(空間参照系管理) “SRS”
- 地名辞典(地名辞典管理) “Gazetteer”
- メタデータ検索 “CSW”

(iii) request

要求名は、” GetCapabilities ” でなければならない。

【書式】

“GetCapabilities”を指定する。

(iv) format

レスポンスの書式を指定する。省略された場合は、” text/xml ” とみなされる。

【書式】

format は、有効な MIME-Type の文字列で定義される。

(v) updateSequence

リクエストがサービスメタデータをキャッシュしていた場合、キャッシュしているメタデータの連番を指定することで、プロバイダとの差異を確認する。サービスメタデータの連番を指定しない場合は、プロバイダが提供可能なバージョンのうち、最新の連番を指定したことと同様の動作となる。リクエストが指定した連番とプロバイダが返す連番などの関係は、以下のとおり。

リクエストが指定する連番	プロバイダが返す連番
指定なし	最新の連番を返す (最新のサービスメタデータを返す)
指定あり (プロバイダ保有の連番と同じの場合)	最新の連番を返す (NULL 空のサービスメタデータを返す)
指定あり (プロバイダ保有の連番以下の場合)	最新の連番を返す (最新のサービスメタデータを返す)
指定あり (プロバイダ保有の連番より大きい場合)	例外処理: 指定した連番は無効

【書式】

サービスメタデータの連番は、整数値で定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
ServiceMetadata 構造体	要求内容に沿ったサービスメタデータ型のデータを送信する

サービスメタデータの内容は、構造体「ServiceMetadata」の型として送信される。ServiceMetadata は、要求するサービスの種類により異なる。以下に、本ガイドラインで規定するサービスの種類に対応する ServiceMetadata 構造体を示す。

サービスの種類	識別子	応答される ServiceMetadata 構造体
地理情報	WMS	WMSServiceMetadata
地名辞典(空間参照系管理)	SRS	SRSServiceMetadata
地名辞典(地名辞典管理)	Gazetteer	GazetteerServiceMetadata
メタデータ検索	CSW	CSWServiceMetadata

5) 例外処理

リクエストしたサービスメタデータの連番が、プロバイダで保有する連番よりも大きい場合など(3-2-5-6.1)に示した InvalidUpdateSequence を利用する)。

4-2-3. 地理情報に関するインタフェース

地理情報に関するインタフェースにおける SOAP と WSDL 利用の実装仕様は、本ガイドラインにおいては特に定めない。

本ガイドラインにおいては、地理情報のインタフェースに関しては、4-1-1 に示すとおり、ISO19128 に基づく POST/GET 利用の実装を推奨する。

4-2-4. 地名辞典(空間参照系管理)に関するインタフェース

4-2-4-1. SearchLRS

1) 機能説明

地理識別子による空間参照系を検索する。空間参照系が1件もヒットしなかった場合は要素数0の配列を返す。

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
String	name	指定しない場合は Null を入力する	名称
GeographicExtent	domainOfValidity	指定しない場合は Null を入力する	地理的範囲
String	theme	指定しない場合は Null を入力する	分類
String	overallOwner	指定しない場合は Null を入力する	管理者
String[]	locationType	必須	関連する場所型の名称 複数指定された場合は AND 検索を行う
AuthResult	authResult	指定しない場合は Null を入力する	認証結果 ユーザ認証を必要としないサービスの場合は null を指定する

(i) name

地理識別子による空間参照系の名称を指定する。

【書式】

特になし。文字列で定義される。

(ii) domainOfValidity

地理識別子による空間参照系が含まれる地理的範囲の情報を指定する。

【書式】

GeographicExtent 構造体を用いて、東西南北の境界座標を定義する。

(iii) theme

地理識別子による空間参照系が持つ分類（主題）を指定する。

【書式】

特になし。文字列で定義される。

(iv) overallOwner

地理識別子による空間参照系の管理者の名称を指定する。

【書式】

特になし。文字列で定義される。

(v) locationType

地理識別子による空間参照系が持つ場所型の名称を指定する。複数指定可能であり、複数指定は配列として指定する。複数指定された場合は、AND 検索となる。

【書式】

場所型の名称（String）の配列として定義される。

(vi) authResult

認証結果を指定する。

ユーザ認証を必要としないサービスの場合は null を指定する。

【書式】

AuthResult 構造体によって定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LRS 構造体の配列	リクエスト内容に応じた地理識別子による空間参照系のリストを送信する

【関連する構造体】

- GeographicExtent 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-4-2. GetLRS

1) 機能説明

地理識別子による地理識別子による空間参照系を取得する

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
String	name	指定しない場合は Null を入力する	名称
AuthResult	authResult	指定しない場合は Null を入力する	認証結果 ユーザ認証を必要としないサービスの場合は null を指定する

(i) name

地理識別子による空間参照系の名称を指定する。

【書式】

特になし。文字列で定義される。

(ii) authResult

認証結果を指定する。

ユーザ認証を必要としないサービスの場合は null を指定する。

【書式】

AuthResult 構造体によって定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LRS 構造体	リクエスト内容に応じた地理識別子による空間参照系を送信する

【関連する構造体】

➤ AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-4-3. SearchLocationType

1) 機能説明

登録されている場所型を検索する。

Irs に地理識別子による空間参照系の名称を指定することで、特定の空間参照系に関連する場所型を検索することが可能である。

また、場所型の Nesting 関連の条件を指定する場合は、parentType または childType に関連する場所型の名称を指定する。

地理的範囲が指定された場合は、場所型が持つ地理的範囲と一部でも重なるものを返す。

条件に合致する場所型が存在しない場合にはサイズ 0 の配列を返す。

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
String	name	指定しない場合は Null を入力する	名称
String	theme	指定しない場合は Null を入力する	分類
PIType	identification	必須	PI タイプ
GeographicExtent	territoryOfUse	指定しない場合は Null を入力する	地理的範囲
String	owner	必須	管理者
String[]	Irs	指定しない場合は Null を入力する	関連する空間参照系の名称
String[]	parentType	指定しない場合は Null を入力する	親の場所型の名称
String[]	childType	指定しない場合は Null を入力する	子の場所型の名称
AuthResult	authResult	指定しない場合は Null を入力する	認証結果

(i) name

場所型の名称を指定する。一部のみの指定も可能である。名称を特定しない場合は null を指定する。

【書式】

場所型の名称は、文字列型によって定義される。

(ii) theme

場所型の分類を指定する。一部のみの指定も可能である。分類を特定しない場合は null を指定する。

【書式】

場所型の分類は、文字列型によって定義される。

(iii) identification

PI タイプの名称を指定する。PI タイプの各要素の値は一部のみの指定が可能。

【書式】

PI タイプの名称は、PIType 型によって定義される。

(iv) territoryOfUse

場所型が含まれる地理的範囲を指定する。

【書式】

地理的範囲は、GeographicExtent 型によって定義される。

(v) owner

場所型の管理者の名称を指定する。

【書式】

管理者の名称は、文字列型によって定義される。

(vi) lrs

場所型が関連する地理識別子による空間参照系の名称を指定する。空間参照系を特に限定しない場合は、null を指定する。複数の名称を指定できるが、複数指定した場合は OR 検索となる。

【書式】

地理識別子による空間参照系の名称は、文字列型の配列として定義される。

(vii) parentType

親子関連の親に相当する場所型の名称を指定する。複数指定可能であるが、親関連の条件を指定しない場合は、null を指定する。

【書式】

親の場所型の名称は、文字列型の配列として定義される。

(viii) childType

親子関連の子に相当する場所型の名称を指定する。複数指定可能であるが、子関連の

条件を指定しない場合は、null を指定する。

【書式】

子の場所型の名称は、文字列型の配列として定義される。

(ix) authResult

認証結果を指定する。ユーザ認証を必要としないサービスの場合は、null を指定する。

【書式】

認証結果は、authResult 構造体により定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LocationType 構造体の配列	リクエスト内容に合致する場所型を、LocationType 構造体として返す。複数の場所型が合致した場合は、LocationType 構造体の配列として返す。

リクエスト時、引数に地理的範囲が指定された場合は、場所型が持つ地理的範囲と一部でも重なるものを返す。

条件に合致する場所型が存在しない場合にはサイズ 0 の配列を返す。

【関連する構造体】

- PIType 構造体
- GeographicExtent 構造体
- LocationType 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-4-4. GetLocationType

1) 機能説明

登録されている場所型を取得する。

2) 利用条件

- ・ 場所型の名称が明らかである場合に利用する。場所型の名称が明らかでない場合は、SearchLocationType 関数を利用する。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
String	name	必須	名称
AuthResult	authResult	指定しない場合は Null を入力する	認証結果

(i) name

場所型の名称を指定する。

【書式】

場所型の名称は、文字列型によって定義される。

(ii) authResult

認証結果を指定する。ユーザ認証を必要としないサービスの場合は、null を指定する。

【書式】

認証結果は、authResult 構造体により定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LocationType 構造体の配列	リクエスト内容に合致する場所型を、LocationType 構造体として返す。

条件に合致する場所型が存在しない場合にはサイズ 0 の配列を返す。

【関連する構造体】

- LocationType 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-4-5. SearchLocationTypeAssociation

1) 機能説明

登録されている場所型関連を検索する。

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
String	childLocationType	parentLocationType を指定しない場合は必須	子となる場所型関連の名称
String	parentLocationType	childLocationType を指定しない場合は必須	親となる場所型関連の名称
AuthResult	authResult	指定しない場合は Null を入力する	認証結果

(i) childLocationType

子となる場所型関連の名称を指定する。parentLocationType を指定しない場合は、必須である。

【書式】

子場所型は、文字列型として定義される。

(ii) parentLocationType

親となる場所型関連の名称を指定する。childLocationType を指定しない場合は、必須である。

【書式】

親場所型は、文字列型として定義される。

(iii) authResult

認証結果を指定する。ユーザ認証を必要としないサービスの場合は、null を指定する。

【書式】

認証結果は、AuthResult 構造体により定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LocationTypeAssociation 構造体	子場所型を指定しない場合は親場所型で指定された場所型のすべての子場所型を返す。 親場所型を指定しない場合は子場所型で指定された場所型のすべての親場所型を返す。

【関連する構造体】

- LocationTypeAssociation 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-5. 地名辞典(地名辞典管理)に関するインタフェース

4-2-5-1. SearchPI

1) 機能説明

登録されている場所インスタンスの PI 部分を検索する。

地理的範囲で地理的境界またはポリゴンを指定した場合は、一部でも地理的範囲が交差する PI を検索する。

条件に合致する PI が存在しない場合はサイズ 0 の配列を返す。

(A) `routeCondition.cascadingLevel > 0` の場合

他の地名辞典管理サービスの `SearchPI` 関数を呼び出す場合は `routeCondition.cascadingLevel` の値を 1 だけ減じ、かつ `routeCondition.routeHistory` の末尾に自サーバを追加しなければならない。また、`routeCondition.routeHistory` で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。

(B) `routeCondition.cascadingLevel = 0` の場合

他の地名辞典管理サービスの `SearchPI` 関数を呼び出すことはできない。また、`routeCondition.routeHistory` で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。

(C) `routeCondition.cascadingLevel < 0` の場合

他の地名辞典管理サービスの `SearchPI` 関数は `routeCondition.routeDefinition` の順に呼び出さなければならない。

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
RouteCondition	routeCondition	指定しない場合は Null を入力する	他サービス呼び出し条件
String	gazetteer	指定しない場合は Null を入力する	地名辞典の名称
String	locationType	指定しない場合は Null を入力する	場所型の名称
String[]	geographicIdentifier	指定しない場合は Null を入力する	地理識別子
GeographicExtent	geographicExtent	指定しない場合は Null を入力する	地理的範囲
TemporalExtent	temporalExtent	指定しない場合は Null を入力する	時間範囲
String	administrator	指定しない場合は Null を入力する	管理者
String[]	searchWord	指定しない場合は Null を入力する	検索語句
AuthResult	authResult	指定しない場合は Null を入力する	認証結果

(i) routeCondition

他サービス呼び出し条件を指定する。

【書式】

他サービス呼び出し条件は、RouteCondition 構造体によって定義される。

(ii) gazetteer

PI が含まれる地名辞典の名称を指定する。地名辞典を特定しない場合は、null を指定する。

【書式】

地名辞典の名称は、文字列型によって定義される。

(iii) locationType

PI が関連する場所型の名称を指定する。場所型を特定しない場合は、null を指定する。

【書式】

場所型の名称は、文字列型によって定義される。

(iv) geographicIdentifier

PI に含まれる地理識別子を指定する。複数指定可能であり、複数指定する場合は地配列として指定する。地理識別子を特定しない場合は、null を指定する。

【書式】

地理識別子は、文字列型の配列によって定義される。

(v) geographicExtent

PI が含まれる地理的範囲を指定する。地理的範囲を限定しない場合は、null を指定する。

【書式】

地理的範囲は、GeographicExtent 構造体により定義される。

(vi) temporalExtent

PI が持つ有効な時間範囲を指定する。時間範囲を限定しない場合は、null を指定する。

【書式】

時間範囲は、TemporalExtent 構造体により定義される。

(vii) administrator

PI の管理者の名称を指定する。管理者を特定しない場合は、null を指定する。

【書式】

管理者の名称は、文字列型により定義される。

(viii) searchWord

検索語句を指定する。検索語句は、場所インスタンスの PI の値または代替地理識別子を対象とした全文検索用フリーワードとして用いられる。語句が複数指定された場合は、地理識別子、または代替地理識別子のいずれかがすべての検索語句を含むものを検索する。

検索語句を設定しない場合は null を指定する。

【書式】

検索語句は、文字列型の配列として定義される。

(ix) authResult

認証結果を指定する。ユーザ認証を必要としないサービスの場合は、null を指定する。

【書式】

認証結果は、AuthResult 構造体により定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
PI 構造体の配列	指定した条件に合致する PI を返す。複数合致した場合は、PI 構造体の配列として返す。

【関連する構造体】

- GeographicExtent 構造体
- TemporalExtent 構造体
- PI 構造体
- RouteCondition 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-5-2. GetLocationInstance

1) 機能説明

登録されている場所インスタンスを取得する。

指定した PI に合致する場所インスタンスを取得する。

地名辞典を特定して場所インスタンスを取得することもできる。

PI が複数指定された場合は、指定された順に対応する場所インスタンスを返す。PI に対応する場所インスタンスが存在しない場合は、配列中の当該要素に null を格納して返す。

条件に合致する場所インスタンスが存在しない場合はサイズ 0 の配列を返す。

(A) routeCondition.cascadingLevel > 0 の場合

他の地名辞典管理サービスの GetLocationInstance 関数を呼び出す場合は routeCondition.cascadingLevel の値を 1 だけ減じ、かつ routeCondition.routeHistory の末尾に自サーバを追加しなければならない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。

(B) routeCondition.cascadingLevel = 0 の場合

他の地名辞典管理サービスの GetLocationInstance 関数を呼び出すことはできない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。

(C) routeCondition.cascadingLevel < 0 の場合

他の地名辞典管理サービスの GetLocationInstance 関数は routeCondition.routeDefinition の順に呼び出さなければならない。

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
RouteCondition	routeCondition	指定しない場合は Null を入力する	他サービス呼び出し条件
String	gazetteer	指定しない場合は Null を入力する	地名辞典の名称
PI[]	pi	指定しない場合は Null を入力する	PI
AuthResult	authResult	指定しない場合は Null を入力する	認証結果

(i) routeCondition

他サービス呼び出し条件を指定する。

【書式】

他サービス呼び出し条件は、RouteCondition 構造体によって定義される。

(ii) gazetteer

PI が含まれる地名辞典の名称を指定する。地名辞典を特定しない場合は、null を指定する。

【書式】

地名辞典の名称は、文字列型によって定義される。

(iii) pi

PI を指定する。複数の PI を指定することができ、その場合は PI の配列を指定する。PI を特定しない場合は、null を指定する。

【書式】

PI は、PI 構造体によって定義される。

(iv) authResult

認証結果を指定する。ユーザ認証を必要としないサービスの場合は、null を指定する。

【書式】

認証結果は、AuthResult 構造体により定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LocationInstance 構造体の配列	検索条件に合致する場所インスタンスを、LocationInstance 構造体の配列として返す。

【関連する構造体】

- PI 構造体
- LocationInstance 構造体
- RouteCondition 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-5-3. SearchLocationInstanceAssociation

1) 機能説明

複数場所インスタンス関連を検索する。

条件に合致する場所インスタンス関連が存在しない場合はサイズ 0 の配列を返す。

(A) routeCondition.cascadingLevel > 0 の場合

他の地名辞典管理サービスの SearchLocationInstanceAssociation 関数を呼び出す場合は routeCondition.cascadingLevel の値を 1 だけ減じ、かつ routeCondition.routeHistory の末尾に自サーバを追加しなければならない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。

(B) routeCondition.cascadingLevel = 0 の場合

他の地名辞典管理サービスの SearchLocationInstanceAssociation 関数を呼び出すことはできない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。

(C) routeCondition.cascadingLevel < 0 の場合

他の地名辞典管理サービスの SearchLocationInstanceAssociation 関数は routeCondition.routeDefinition の順に呼び出さなければならない。

Nesting 関連を検索する場合は source を child に、target を parent に読み替える。

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
RouteCondition	routeCondition	必須	他サービス呼び出し条件
String	sourceGazetteer	指定しない場合は Null を入力する	関連元場所インスタンスの地名辞典の名称
String	sourceGeographicIdentifier	指定しない場合は Null を入力する	関連元場所インスタンスの識別子
String	targetGazetteer	指定しない場合は Null を入力する	関連先場所インスタンスの地名辞典の名称
String	targetGeographicIdentifier	指定しない場合は Null を入力する	関連先場所インスタンスの地理識別子
String[]	phase	必須	位相の名称
AuthResult	authResult	指定しない場合は Null を入力する	認証結果

(i) routeCondition

他サービス呼び出し条件を指定する。

【書式】

他サービス呼び出し条件は、RouteCondition 構造体によって定義される。

(ii) sourceGazetteer

関連元場所インスタンスの地名辞典の名称を指定する。関連元場所インスタンスの地名辞典を特定しない場合は null を指定する。

【書式】

関連元場所インスタンスの地名辞典の名称は、文字列型として定義される。

(iii) sourceGeographicIdentifier

関連元場所インスタンスが含む地理識別子を指定する。関連元場所インスタンスを特定しない場合は null を指定する。

【書式】

関連元場所インスタンスが含む地理識別子は、文字列型として定義される。

(iv) targetGazetteer

関連先場所インスタンスの地名辞典の名称を指定する。関連先場所インスタンスの地名辞典を特定しない場合は null を指定する。

【書式】

関連先場所インスタンスの地名辞典の名称は、文字列型として定義される。

(v) targetGeographicIdentifier

関連先場所インスタンスが含む地理識別子を指定する。関連先場所インスタンスを特定しない場合は null を指定する。

【書式】

関連先場所インスタンスが含む地理識別子は、文字列型として定義される。

(vi) phase

位相の名称を指定する。個々の値には、

“ Nesting ” または
“ Equivalent ” を指定する。
複数指定した場合は OR 検索を行う。

【書式】

位相の名称は複数指定することができ、文字列型の配列として定義される。

(vii) authResult

認証結果を指定する。ユーザ認証を必要としないサービスの場合は、null を指定する。

【書式】

認証結果は、AuthResult 構造体により定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LocationInstanceAssociation 構造体の配列	検索条件に合致する場所インスタンス関連を、LocationInstanceAssociation 構造体の配列として返す。 検索条件に合致する場所インスタンス関連が存在しない場合は、サイズ0の配列を返す。

【関連する構造体】

- LocationInstanceAssociation 構造体
- RouteCondition 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-6. メタデータ検索に関するインタフェース

メタデータ検索に関するインタフェースにおける SOAP と WSDL 利用の実装仕様は、OGC Catalogue Service Specification において規定されている、HTTP を通信プロトコルとした実装仕様である Catalogue Services for the Web (CSW) に準ずる。SOAP と WSDL 利用のためのインタフェースを実装する場合には、CSW のレスポンスの規則、リクエストパラメータの規則、各関数の記述規則等に従うものとする。

本ガイドラインで規定する CSW のプロトコルバージョンは、「2.0.0」とする。

5. 関数仕様作成に関する取り決め

5-1. 関数仕様作成に関する取り決め

アプリケーションプログラムを開発する場合、そのアプリケーションプログラムで使用する関数は、本ガイドラインの「標準インタフェース関数仕様」で定める関数とする。

アプリケーションプログラムで使用する関数が、本ガイドラインの「標準インタフェース関数仕様」で定める関数に該当しない場合には、当該アプリケーションプログラムの開発者は、使用する関数について関数仕様を作成する。

関数仕様は、以下の2種類を作成する。

- (1) 使用する関数一覧
- (2) 各関数の説明文書

【解説】

アプリケーションプログラムの開発者は、当該アプリケーションプログラムで使用する関数が本ガイドラインの「標準インタフェース関数仕様」で定める関数に該当しない場合に、関数の概要を説明する「使用する関数一覧」および関数の詳細を説明する「各関数の説明文書」の2種類を作成しなくてはならない。

(1) 使用する関数の一覧

アプリケーションプログラムが使用する関数の一覧を作成する。一覧表には、使用する各関数について「機能分類」「機能」「内容」「関数名」を記載しなくてはならない(表 5-1)。記載例を表 5-2 に示す。

表 5-1 使用する関数一覧の記載項目

項目	定義	データ型	要求度/条件	最大記述回数
関数	アプリケーションプログラムで使用する関数	クラス	必須	N
機能分類	機能について、それをいくつかのまとまりでとりまとめたもの	文字列	任意	1
機能	関数が行う処理を簡潔に説明したもの	文字列	必須	1
内容	関数が行う処理内容の概要	文字列	必須	1
関数名	関数の名前	文字列	必須	1

表 5-2 使用する関数一覧の記載例

機能分類	機能	内容	関数名
初期設定	サービスメタデータ取得	利用可能な操作の情報を取得する	GetCapabilities
...

(2) 各関数の説明文書

一覧として作成したアプリケーションプログラムが使用する各関数について、以下の項目を記載した説明文書を作成する。

- 関数名
- 説明
- 引数
- 戻り値
- 関連項目

記載事項の詳細を表 5-3 に示す。関数の引数および戻り値の詳細は、表 5-4 および表 5-5 のとおりである。記載例を表 5-6 に示す。

表 5-3 各関数の説明文書記載事項

項目	定義	データ型	要求度/条件	最大記述回数
関数説明文書	アプリケーションプログラムで使用する各関数についての説明	クラス	必須	N
関数名	関数の名称	文字列	必須	1
説明	その関数が行う処理内容の概要	文字列	必須	1
引数	関数を呼び出すときに相手に渡す値。呼び出された側では、この引数に応じて処理を行う	クラス 表 5-4	必須	N
戻り値	関数が処理を終了し、処理の結果として返す値	クラス 表 5-5	必須	1
関連項目	当該関数で参照する他の関数	文字列	任意(引数と戻り値で、構造体を用いる場合)	N

表 5-4 各関数の説明文書記載事項（引数）

項目	定義	データ型	要求度/条件	最大記述回数
引数	関数を呼び出すときに相手に渡す値。呼び出された側では、この引数に応じて処理を行う	クラス	必須	N
引数の型	引数のデータ型	文字列	必須	1
引数の名前	引数の名称	文字列	必須	1
引数の説明	引数が意味する内容、引数の定義域	文字列	必須	1
引数の要求度/条件	当該関数に対し、当該引数が必須であるか任意であるかの区別。また、その場合の条件。	文字列	必須	1

表 5-5 各関数の説明文書記載事項（戻り値）

項目	定義	データ型	要求度/条件	最大記述回数
戻り値	関数が処理を終了し、処理の結果として返す値	クラス	必須	1
戻り値の型	戻り値の型	文字列	必須	1
戻り値の説明	戻り値が意味する内容	文字列	必須	1

表 5-6 関数の説明文書の記載例

名称	GetCapabilities		
機能	サービスメタデータを取得する		
引数	型	名前	説明
	String	version	準拠する標準インターフェースのバージョン番号 該当するバージョンのサービスメタデータのみを取得する 指定なしの場合は、最新バージョン
	String	service	サービスタイプ 要求するサービスの識別子 例) 地理情報 “WMS” 地名辞典(空間参照系管理) “SRS” 地名辞典(地名辞典管理) “Gazetteer” メタデータ検索 “CSW”
	String	request	要求名 ”GetCapabilities”でなければならない
	String	format	サービスメタデータの出カフォーマット 応答をどのようなフォーマット(MIME Type)で返すかを指定する 省略された場合は、”text/xml”とみなされる
	String	updateSequence	サービスメタデータの連番 サービス呼び出し側がこの連番に該当するサービスメタデータを キャッシュしている場合に指定する
戻り値	ServiceMetadata	サービスメタデータ	
関連項目	ServiceMetadata 構造体 (※引数”service”において指定したサービスタイプにより、構造体の内容は異なる)		

5-2. 関数と引数の命名規則

関数仕様を作成する場合の関数と引数の命名規則を、以下のとおり取り決める。

関数の命名規則

- 識別子の名前は英語を基本とする。
- 動詞または動詞句とする。
- 識別子の最初の文字と、後に続いて連結されている各単語の最初の文字を大文字にする。

引数の命名規則

- 識別子の名前は英語を基本とする。
- 識別子の最初の文字は小文字にし、後に続いて連結されている各単語の最初の文字を大文字とする(Camel 形式)。
- 原則として、引数の型を説明する名前ではなく、引数の意味を説明する名前を使用する。

【解説】

アプリケーションプログラムの開発者は、当該アプリケーションプログラムで使用する関数が本ガイドラインの「標準インタフェース関数仕様」で定める関数(第3章参照)に該当しない場合に、使用する関数について関数仕様を作成する(5-1節参照)。作成する関数仕様は、本ガイドラインで定める関数仕様と同様に、他のアプリケーション開発者が利用する。したがってアプリケーション開発者が混乱しないよう、その名称を、同じ規則で命名する必要がある。従って、関数と引数についての命名規則を以下のとおり取り決める。

関数、引数とも、命名は英語を基本とする。

関数については、動詞または動詞句とする。特に、処理内容が相対的な関係にある関数の命名にあたっては、以下の英語の相対性に留意する。

- Add / Remove
- Insert / Delete
- Get / Set
- Start / Stop
- Begin / End
- Send / Receive
- First / Last
- Get / Release
- Put / Get

- **Up / Down**
- **Show / Hide**
- **Source / Target**
- **Open / Close**
- **Source / Destination**
- **Increment / Decrement**
- **Lock / Unlock**
- **Old / New**
- **Next / Previous**

6. 実装仕様作成に関する取り決め

6-1. 実装仕様作成に関する取り決め

アプリケーションプログラムで使用する関数が、本ガイドラインの「標準インタフェース関数仕様」で定める関数に該当しない場合には、当該アプリケーションプログラムの開発者は、「標準インタフェース関数仕様作成に関する取り決め」に従い作成した関数仕様に対応する実装仕様を作成する。

この場合の実装は、HTTP による SOAP と WSDL を利用した実装とする。

実装仕様は、以下の 2 種類を作成する。

- (1) 実装仕様の概要
- (2) WSDL による記述文書

【解説】

アプリケーションプログラムの開発者は、当該アプリケーションプログラムで使用する関数が、本ガイドラインの「標準インタフェース関数仕様」で定める関数（第 3 章参照）に該当しない場合には、「標準インタフェース関数仕様作成に関する取り決め」に従い関数仕様を作成する（5 章参照）。当該アプリケーションプログラムの開発者は、さらに作成した関数仕様に対応する実装仕様を作成しなければならない。

実装仕様は、人間にとって可読性の高い「実装仕様の概要」および機械にとって可読性の高い「WSDL による説明文書」の 2 種類からなる。作成する実装仕様の詳細を次に示す。

(1) 実装仕様の概要

機能の説明や処理の内容について、文書形式により作成する。作成する実装仕様は、本ガイドラインで示す実装仕様と同様の書式とする。実装仕様の概要として記載しなければならない項目とその内容を、表 6-1 に示す。

表 6-1 実装仕様の概要の記載項目

項目	定義	データ型	要求度/条件	最大記述回数
関数名	実装仕様を記述する関数の名前	文字列	必須	1
機能説明	関数の処理内容の詳細説明	文字列	必須	1
利用条件	関数を利用する場合に必要な条件	文字列	必須	1
リクエストの内容	リクエストに使用するパラメータの説明	文字列	必須	1
レスポンスの内容	レスポンスで受け渡される情報の説明	文字列	必須	1
例外処理	無効なリクエストが発行された場合やサービスが実装されていない場合などの例外処理の発生時の説明と、その処理内容	文字列	必須	1

(2) WSDL による記述文書

SOAP でのメッセージの受け渡しを考慮し、WSDL の仕様に基づき WSDL 文書を作成する。

WSDL 文書の作成単位は、アプリケーションプログラム単位とする。

7. 標準インタフェースの運用に関する取り決め

7-1. インタフェース公開・管理方法

アプリケーションプログラムの開発者は、開発するアプリケーションプログラムが他のアプリケーションプログラムから接続できるよう、インタフェースを公開する。公開するインタフェースは、本ガイドラインに基づき作成する。

アプリケーションプログラムの開発者は、開発するアプリケーションプログラムについて、アプリケーションプログラムのインタフェースに関する以下の2種類の情報を記載した文書を作成する。

(1) アプリケーションプログラムの内容

- アプリケーションプログラムの目的と、機能およびデータの概要
- 利用にあたっての制限や条件
- 接続方法
- 利用許可・利用申請のための問合せ先

(2) アプリケーションプログラムで利用できる関数等の技術的な仕様

これらの文書は、別途標準インタフェースの管理機関において公開する。

【解説】

本ガイドラインに従って開発するアプリケーションプログラムを有効に利用するために、アプリケーションプログラムの開発者は、開発するアプリケーションプログラムが他のアプリケーションプログラムから接続できるよう、本ガイドラインに基づくインタフェースを実装し、それを公開する。

公開するインタフェースに関する情報は、以下の用途に利用する。

- アプリケーションプログラムを利用したい機関が、利用可能なアプリケーションプログラムの有無を検索する。
- 利用にあたって、アプリケーションプログラムに関する概要ならびに開発に必要な技術情報を入手する。

アプリケーションプログラムの開発者は、公開するインタフェースを他者が利用するために、アプリケーションプログラムのインタフェースに関する情報を文書として、アプリケーションプログラムの概要を説明する「アプリケーションプログラムの内容」およびアプリケーションプログラムの技術的な内容を説明する「アプリケーションプログラムで利用できる関数等の技術的な仕様」の2種類を作成しなければならない。作成した文書は、インタフェースを利用しようとする機関がその情報を入手できるよう公開する。

公開する情報は、表 7-1 に示す情報とする。その仕様は、公開・管理仕様（8章）に示す。

表 7-1 公開する情報

公開する情報		説明
(1) アプリケーションプログラムの内容	アプリケーションプログラムの目的と、機能およびデータの概要	<ul style="list-style-type: none"> ■ アプリケーションプログラム作成の目的 ■ アプリケーションプログラムで提供可能な機能 ■ アプリケーションプログラムで取り扱うデータの種類、データの範囲に関する情報
	利用にあたっての制限や条件	<ul style="list-style-type: none"> ■ プログラムやデータに関する利用上の制限 <ul style="list-style-type: none"> □ 著作権保護に関する情報 □ 使用料に関する情報 ■ アプリケーションプログラムを利用するための物理的な環境に関する情報 <ul style="list-style-type: none"> □ 利用可能な時間に関する情報 □ 利用可能な期間に関する情報 □ 利用可能なデータの範囲に関する情報 □ ネットワーク環境に関する情報
	接続方法	<ul style="list-style-type: none"> ■ 接続ポイントに関する情報(URL など)
	利用許可・利用申請のための問合せ先	<ul style="list-style-type: none"> ■ アプリケーションプログラムを利用するにあたり、利用許可や利用申請を行うために必要となる問合せ先や問合せ方法に関する情報 <ul style="list-style-type: none"> □ アプリケーションプログラムの管理者の組織名称・担当者役職・問合せ方法・問合せ可能時間・電話番号・住所等 □ アプリケーションプログラムの開発者の組織名称・担当者役職・問合せ方法・問合せ可能時間・電話番号・住所等 □ アプリケーションプログラムやデータの著作権者の組織名称・担当者役職・問合せ方法・問合せ可能時間・電話番号・住所等
(2) アプリケーションプログラムで利用できる関数等の技術的な仕様	<ul style="list-style-type: none"> ■ アプリケーションプログラムが対応している関数・引数・戻り値に関する情報(関数仕様) ■ アプリケーションプログラム間でそれらがやり取りされる形式に関する情報(実装仕様) ■ 取得可能なデータの情報 	

7-2. インタフェース接続方式

他のアプリケーションプログラムと接続する場合は、以下の手順で接続する。

- (1) 標準インタフェース登録機関において、利用可能なアプリケーションプログラムを検索する。
- (2) 利用するアプリケーションプログラムについて、そのアプリケーションプログラムの管理者に利用の申請を行う。必要に応じて、利用するための各種条件の処理を行う。
- (3) 利用に必要な関数仕様・実装仕様等の技術資料を入手し、アプリケーションプログラムを利用するためのインタフェースを実装する。
- (4) アプリケーションプログラムに接続し、利用する。

アプリケーションプログラム間の接続は、本ガイドラインで定めるインタフェースを介して行う。

接続に用いる通信プロトコルは、HTTP とし、以下のうちいずれかの方式とする。接続するアプリケーションプログラムの実装仕様に従い、接続を行う。

方式1) POST/GET による方法

方式2) SOAP による方法

【解説】

他のアプリケーションプログラムと接続する場合には、まず既に登録してあるアプリケーションプログラムに関する情報を検索し、利用可能なアプリケーションプログラムが存在するかどうか確認を行う。利用可能なアプリケーションプログラムが見つかった場合には、あわせて登録してあるアプリケーションプログラムの管理者情報を基に、そのアプリケーションプログラムの管理者に対して接続・利用申請を行う。利用にあたっての条件等が解決し、利用に問題がなければ、公開されている技術的な仕様を基に、接続のためのインタフェースを実装する。

なお、アプリケーションプログラム間の接続に用いる通信プロトコルは HTTP とする。アプリケーションプログラムの開発者は、方式1あるいは方式2を採用し、アプリケーションプログラムを接続する必要がある。アプリケーションプログラムの開発者は、以下のいずれかの方式で実装する。

方式1) POST/GET による方法

方式2) SOAP による方法

8. 標準インタフェース公開・管理仕様

標準インタフェース公開・管理仕様として、以下の 2 種類の文書を定義する。

- 公開・管理のためのアプリケーションプログラムに関する文書
- 公開・管理のためのアプリケーションプログラムで利用できる関数等の技術的文書

【解説】

標準インタフェース公開・管理仕様では、以下の 2 種類の文書を定義する。アプリケーションプログラムの開発者は、公開するインタフェースを他者が利用するために、これらの 2 種類を作成しなければならない。

- 公開・管理のためのアプリケーションプログラムに関する文書
- 公開・管理のためのアプリケーションプログラムで利用できる関数等の技術的文書

「公開・管理のためのアプリケーションプログラムに関する文書」は、アプリケーションプログラムの目的、機能、取り扱うデータ、利用時の制限や条件、接続方法、および問合せ先について、その概要を明らかにするための文書である。

「公開・管理のためのアプリケーションプログラムで利用できる関数等の技術的文書」は、アプリケーションプログラムがどのようなインタフェースを公開しているのか、具体的な関数とそれに関する引数等の条件を明らかにするための文書である。

1 つのインタフェース公開情報には 1 つの実装方式のみを記述する。たとえば、HTTP POST/GET と SOAP (セッションあり、セッションなし) をすべて実装するアプリケーションプログラムは、合計 3 つのインタフェース公開情報を作成する。

8-1. アプリケーションプログラムの内容に関する公開・管理仕様

公開・管理のためのアプリケーションプログラムの内容に関する文書は、以下の内容を含む。

- (1) アプリケーションプログラムの目的と、機能およびデータの概要
- (2) 利用にあたっての制限や条件
- (3) 接続方法
- (4) 利用許可・利用申請のための問合せ先

【解説】

上述 4 項目の詳細を以下の (1) から (4) で示す。この 4 つの項目は公開・管理仕様として原則必須であるが、(4) についてはアプリケーションプログラムの開発者のみ必須である。

(1) アプリケーションプログラムの目的と、機能およびデータの概要

「アプリケーションプログラムの目的と、機能およびデータの概要」として記載する項目は、「名称」「概要」「目的」「データの種類」「データの範囲」である。詳細を表 8-1 に示す。

表 8-1 アプリケーションプログラムの目的と機能およびデータの概要記載項目

項目	定義	データ型	要求度/ 条件	最大記述 回数
アプリケーションプログラムの目的と、機能およびデータの概要	アプリケーションプログラムで実現可能な機能等の概要	クラス	必須	1
名称	アプリケーションプログラムの名称	文字列	必須	1
概要	アプリケーションプログラムの機能の概要	文字列	必須	1
目的	アプリケーションプログラムの目的	文字列	必須	1
データの種類	アプリケーションプログラムで取り扱うデータの種類	文字列	必須	N
データの範囲	アプリケーションプログラムで取り扱うデータの空間範囲	文字列	必須	1

(2) 利用にあたっての制限や条件

「利用にあたっての制限や条件」として記載する項目は、「サービス開始時期」「サービス終了予定時期」「運用時間」「著作権」「費用」である。これらの詳細を、表 8-2 に示す。

表 8-2 利用にあたっての制限や条件記載項目

項目	定義	データ型	要求度/ 条件	最大記述 回数
利用にあたっての制限や条件	アプリケーションプログラムを利用するにあたっての制限事項や条件	クラス	必須	1
サービス開始時期	アプリケーションプログラムの利用を開始する時期	文字列	任意	1
サービス終了予定時期	アプリケーションプログラムの利用を終了する時期(予定)	文字列	任意	1
運用時間	アプリケーションプログラムの利用可能な時間	文字列	必須	N
著作権	アプリケーションプログラムとアプリケーションプログラムで取り扱うデータの著作権	文字列	必須	1
費用	アプリケーションプログラムの利用に必要な費用(プログラムやデータの使用料など)	文字列	任意	1

(3) 接続方法

「接続方法」として記載する項目は、「URL」である。詳細を表 8-3 に示す。

表 8-3 接続方法記載項目

項目	定義	データ型	要求度/ 条件	最大記述 回数
接続方法	接続に必要な URL	クラス	必須	1
URL	接続先の URL アドレス	文字列	必須	1

(4) 利用許可・利用申請のための問合せ先

「利用許可・利用申請のための問合せ先」として記載する項目は、「問合せ区分」「問合せ先名称」「住所」「電話番号」「ファックス番号」「電子メールアドレス」および「問合せ条件」である。詳細を表 8-4 に示す。

表 8-4 利用許可・利用申請のための問合せ先記載項目

項目	定義	データ型	要求度/条件	最大記述回数
利用許可・利用申請のための問合せ先	アプリケーションプログラムの管理者等の問合せ先情報	クラス	アプリケーションプログラムの管理者は必須	N
問合せ先区分	問合せ先の区分 以下のいずれかを記入する。 <ul style="list-style-type: none"> ✓ アプリケーションプログラムの管理者 ✓ アプリケーションプログラムの開発者 ✓ 著作権者(著作権者の場合には、何に対して著作権を保持しているか記述する。 	文字列	必須	1
問合せ先名称	管理者や開発者等の組織名、または個人名	文字列	必須	1
住所	問合せ先の住所	文字列	任意	1
電話番号	問合せ先の電話番号	文字列	必須	N
ファックス番号	問合せ先のファックス番号	文字列	任意	N
電子メールアドレス	問合せ先の電子メールアドレス	文字列	任意	N
問合せ条件	問合せに対応可能な時間、曜日	文字列	任意	N

8-2. アプリケーションプログラムで利用できる関数等の技術的な公開・管理仕様

アプリケーションプログラムで利用できる関数等の技術的な公開・管理仕様として、以下の文書を定義する。

- (1) アプリケーションプログラムを開発する際に引用したガイドライン
- (2) 関数仕様
 - 関数一覧
 - 各関数の説明文書
- (3) 実装仕様
 - 実装仕様文書形式
 - WSDL による記述文書
- (4) 取得可能なデータ

【解説】

開発するアプリケーションプログラムで使用するインタフェースには、使用する関数の種類に応じて次のような3つの場合分けがある。

- 1) 本ガイドラインの関数仕様には示されている関数のみを使用する場合
- 2) 本ガイドラインの関数仕様には示されている関数以外の関数を使用する場合（使用する関数すべてが、本ガイドラインの関数仕様には示されている関数に該当しない場合）
- 3) 本ガイドラインの関数仕様には示されている関数に加え、それ以外の関数も使用する場合（使用する関数の一部が、本ガイドラインの関数仕様には示されている関数に該当しない場合）

各場面において作成する関数等の技術的な公開・管理仕様を以下に示す。

- 1) 本ガイドラインの関数仕様には示されている関数のみを使用する場合

開発するアプリケーションプログラムで使用するインタフェースが、本ガイドラインの関数仕様には示されている関数のみの場合、「関数等の技術的な公開・管理仕様」として以下の4文書を作成しなければならない。

1. アプリケーションプログラムを開発する際に引用したガイドライン記載項目を、表 8-5 に示す。
2. 使用した関数の一覧記載項目を、表 8-6 に示す。
3. WSDL による記述文書

アプリケーションプログラム使用する関数についての WSDL 文書を作成する。WSDL 文書は、当該アプリケーションプログラムに対し、1つのファイルとする。

4. 取得可能なデータの一覧

記載項目を、表 8-7 に示す。

「関数等の技術的な公開・管理仕様」の記載例を表 8-8 に示す。

表 8-5 記載項目一覧(1)

項目	定義	データ型	要求度/条件	最大記述回数
ガイドライン	アプリケーションプログラムを開発する際に引用した「標準インタフェースガイドライン」	クラス	必須	1
名称	ガイドラインの名称	文字列	必須	1
版	ガイドラインの版	文字列	必須	1
年月	ガイドラインの版の発行年月	文字列	必須	1

表 8-6 記載項目一覧(2)

項目	定義	データ型	要求度/条件	最大記述回数
関数	アプリケーションプログラムを開発する際に引用した「標準インタフェースガイドライン」の関数仕様に記載されている関数で、当該アプリケーションプログラムで利用できる関数	クラス	必須	N
機能分類	「標準インタフェースガイドライン」の関数仕様に記載されている当該関数の機能分類	文字列	任意	1
機能	「標準インタフェースガイドライン」の関数仕様に記載されている当該関数の機能	文字列	必須	1
内容	「標準インタフェースガイドライン」の関数仕様に記載されている当該関数の内容	文字列	必須	1
関数名	「標準インタフェースガイドライン」の関数仕様に記載されている当該関数の関数名	文字列	必須	1

表 8-7 記載項目一覧 (3)

項目	定義	データ型	要求度/条件	最大記述回数
データ	開発したアプリケーションプログラムで、「標準インタフェースガイドライン」に基づく関数により取得可能なデータ	クラス	必須	N
名称	データの名称	文字列	必須	1
説明	データの定義、取得基準等の説明	文字列	必須	1
内部名称	当該データの、データベース上の名称	文字列	必須	1
属性	各データ項目に対応する属性	クラス	必須	N
名称	属性の名称	文字列	必須	1
内部名称	当該属性の、データベース上の名称	文字列	必須	1

表 8-8 公開・管理仕様作成例

準拠したガイドラインの名称	標準インタフェースガイドライン
準拠したガイドラインの版	第 1.0 版
準拠したガイドラインの年月日	平成 19 年 3 月 XX 日

機能分類	機能	内容	関数名
初期設定	サービスメタデータ取得	利用可能な操作の情報を取得する	GetCapabilities
...

データ	説明	内部名称	属性 1	属性 1 内部名称	属性 2	属性 2 内部名称
メタデータ ID	カタログサービスを利用して得られるメタデータレコードの ID	recordId	ID	metadataId	名称	name
メタデータスキーマ	カタログサービスを利用して得られるメタデータのスキーマ	schema	スキーマ 名称	recordSchema	—	—
メタデータ要素	カタログサービスを利用して得られるメタデータが持つ要素の内容	elementSet	要素 名称	name	要素の 値	value

2) 開発するアプリケーションプログラムで使用するインタフェースが、本ガイドラインの関数仕様に示されていない場合

開発するアプリケーションプログラムで使用するインタフェースが、本ガイドラインの関数仕様に示されていない場合、「関数等の技術的な公開・管理仕様」として以下の 6 文書を作成しなければならない。

1. アプリケーションプログラムを開発する際に引用したガイドライン
記載項目を、表 8-5 に示す。
2. 関数一覧
本ガイドラインの「標準インタフェース関数仕様作成に関する取り決め」(5 章)に従い、関数一覧を作成する。
3. 各関数の説明文書
本ガイドラインの「標準インタフェース関数仕様作成に関する取り決め」(5 章)に従い、各関数の説明文書を作成する。
4. 実装仕様の概要
本ガイドラインの「標準インタフェース実装仕様作成に関する取り決め」(6 章)に従い、各関数について実装仕様の概要を作成する。
5. WSDL による記述文書
アプリケーションプログラム使用する関数についての WSDL 文書を作成する。WSDL 文書は、当該アプリケーションプログラムに対し、1 つのファイルとする。
6. 取得可能なデータの一覧
記載項目を、表 8-7 に示す。

なお、開発するアプリケーションプログラムで使用するインタフェースが、本ガイドラインの関数仕様に示されているかどうかの判断には注意を要する。

- 3) 開発するアプリケーションプログラムで使用するインタフェースが、本ガイドラインの関数仕様に示されている関数と、それ以外の関数の両方存在する場合

開発するアプリケーションプログラムで使用するインタフェースが、本ガイドラインの関数仕様に示されている関数と、それ以外の関数の両方存在する場合、以下を作成する。

1. アプリケーションプログラムを開発する際に引用したガイドライン
記載項目を、表 8-5 に示す。
2. 関数一覧
本ガイドラインの関数仕様に示されている関数と、それ以外の関数とを
区別して、作成する。
本ガイドラインの関数仕様に示されている関数については、表 8-6 に
示す項目とする。
本ガイドラインの関数仕様に示されていない関数については、本ガイド
ラインの「標準インタフェース関数仕様作成に関する取り決め」(5章)に
従い、関数一覧を作成する。
3. 各関数の説明文書
本ガイドラインの関数仕様に示されていない関数についてのみ、本ガイ
ドラインの「標準インタフェース関数仕様作成に関する取り決め」(5章)
に従い、各関数の説明文書を作成する。
本ガイドラインの関数仕様に示されている関数については作成しない。
4. 実装仕様の概要
本ガイドラインの関数仕様に示されていない関数についてのみ、本ガイ
ドラインの「標準インタフェース実装仕様作成に関する取り決め」(6章)
に従い、各関数について実装仕様の概要を作成する。
本ガイドラインの関数仕様に示されている関数については作成しない。
5. WSDL による記述文書
本ガイドラインの関数仕様に示されている関数と、それ以外の関数とを
まとめて、アプリケーションプログラム使用する関数として WSDL 文書
を作成する。WSDL 文書は、当該アプリケーションプログラムに対し、1
つのファイルとする。
6. 取得可能なデータの一覧
記載項目を、表 8-7 に示す。

1. 参考とすべきインタフェース

本ガイドラインでは採用しなかったが、本ガイドラインにおいて定めた内容では情報共有や機能の実現が実現が不可能と思われる場合に関して、以下に示す2つのインタフェースの利用を推奨する。

1-1. 地物単位で地理情報を取得するインタフェース(参考)

本ガイドラインで地理情報に関するインタフェースとして採用した ISO19128 Web Map Server Interface (以下、WMS) は、地理情報を「地図画像」として共有するためのインタフェースである。一方で、地物単位で地理情報を共有するためのインタフェースとして、ISO19142 Web Feature Service (以下、WFS) がある。WFS は、現在 ISO において標準化作業が進行中であるが、平成 19 年 3 月現在で国際規格として制定されていないため、本ガイドラインにおいては参考扱いとする。

1-1-1. 関数仕様

以下に、WFS において規定されているインタフェース仕様の一覧を示す。なお、WFS は通信プロトコルとして HTTP を、データ送受信の手法として POST/GET メソッドの利用を原則としている。

附-表 1-1 WFS のインタフェース

サービス分類	機能分類	機能	内容	操作(関数)
GIS	初期設定	サービスメタデータ取得	利用可能な操作の情報を取得する	GetCapabilities
		地物定義情報取得	地物の定義情報を取得する	DescribeFeatureType
	検索	地物検索	地物の情報を取得する	GetFeature
		地物検索とロック	地物の情報を取得すると共に、編集のために地物をロックする	GetFeatureWithLock
	編集	地物編集ロック	地物に対して編集ロックをかけて他ユーザによる編集を禁止する	LockFeature
		地物編集	地物を追加・編集・削除する	Transaction

1-1-2. 詳細情報の入手先

WFS は、元来は OGC 実装仕様として制定されたものであり、それを基にして ISO により国際規格化の作業が行われている。しかしながら、ISO において作業中の内容に関しては、一般に公開されていない。ISO での国際規格化の元となる OGC 実装仕様に関しては、OGC の Web サイトより入手可能である。同実装仕様は、ISO において国際規格化されるものと詳細は異なる可能性があるが、参考として同実装仕様の URL を以下に示す。WFS の詳細に関しては、以下に示す URL にある文書を参照されたい。

<http://www.opengeospatial.org/standards/wfs> (平成 19 年 8 月時点)

なお、附-表 1-1 で示した WFS インタフェースは、OGC 実装仕様 (OpenGIS® Web Feature Service (WFS) Implementation Specification Version 1.1) である。

1-2. SOAP による地理情報等の標準インタフェース(参考)

本ガイドラインで地理情報に関するインタフェースとして採用した ISO19128 Web Map Server Interface (以下、WMS) は、HTTP プロトコルの POST/GET による接続方式のインタフェース仕様を規定している。

一方で、本ガイドラインでも一部を引用している「河川 GIS・河川アプリケーション標準インタフェースガイドライン(以下、河川インタフェース)」では、地理情報のより高度な利用を目的とし、SOAP による接続方式を採用した地理情報のインタフェースを定めている。河川インタフェースにおいて定めた SOAP による地理情報のインタフェースは、WMS や類似規格である WFS を参考とし、独自にインタフェース仕様を定めたものである。

河川インタフェースにおいて定めている SOAP による地理情報のインタフェースは、ISO 等の標準規格とは異なるものの、河川分野に限らず、汎用的な地理情報のインタフェースとして位置づけられる。したがって、本ガイドラインでは、ISO や OGC において策定されたインタフェースに次ぐ標準インタフェースとして河川インタフェースを位置づけ、参考扱いとして SOAP による地理情報のインタフェースを 1-2-1 に提示する。

1-2-1. 関数仕様

SOAP による地理情報のインタフェースにおいては、セッション管理を行うものを行わないものの2種類のインタフェースを定義する。

「セッション」とは、ユーザが行う一連の行動のまとまりのことである。

「セッション管理ありのインタフェース」とは、ユーザの一連の行動をまとまりとして認識するインタフェースのことである。「セッション管理なしのインタフェース」とは、ユーザの一連の行動をまとまりとして認識しないインタフェースのことである。

セッションがない場合は、ユーザが前の行動の結果を用いて次の行動を起こす場面において、次の行動を実現するインタフェースを呼び出す際に、前の行動の結果を引数として渡さなければならない。それに対して、セッションがある場合は、前の行動と次の行動を同じユーザのものとして認識できるため、前の行動の結果をインタフェース内部に保持しておくことが可能となる。したがって、次の行動を実現するインタフェースを呼び出す際に、前の行動の結果を引数として渡す必要がない。その結果、セッションがある場合インタフェースの引数を省略できるため、ネットワークのトラフィックを削減することができる。

セッション管理の仕組みやトランザクション管理の仕組みを実装しない場合には、「セッション管理なしのインタフェース」の関数仕様に従う。セッション管理の仕組みやトランザクション管理の仕組みを実装するもしくはすでに実装している場合には、そのインタフェースは「セッション管理ありのインタフェース」の関数仕様に従う。

地図の表示レイヤの名称やスタイルなどの指定情報は、レイヤの数に比例して多くなる可能性があるため、省略の対象とすることでインタフェースの性能向上が期待できる。

1-2-1-1. SOAPとWSDLのための関数仕様(セッション管理なし)

以下に、セッション管理なしの場合の SOAP による地理情報のインタフェースの関数仕様の一覧を示す。

附-表 1-2 SOAP による地理情報のインタフェース (セッション管理なし)

サービス分類	機能分類	機能	内容	SOAP 1.1 セッション管理なし
GIS	初期設定	サービスメタデータ取得	利用可能な操作の情報を取得する	GetCapabilities
		データ範囲取得	全データの座標範囲を取得する	GetExtent
		地物定義情報取得	地物の定義情報を取得する	GetFeatureClassInfo
	検索	ID 指定の地物検索	地物の ID を指定して検索し、発見された地物の情報を取得する	GetFeature
		点座標指定の地物検索	点の座標を指定して検索し、発見された地物の情報を取得する	(GetFeature)
		面指定の地物検索	面の座標を指定して検索し、発見された地物の情報を取得する	(GetFeature)
		属性指定の地物検索	地物の属性に条件を設定して検索し、発見された地物の情報を取得する	(GetFeature)
		属性コードリスト取得	コードリスト型の属性項目が取りうる値のリストを取得する	GetCodeList
	編集	地物編集ロック	地物に対して編集ロックをかけて他ユーザによる編集を禁止する	LockFeature
		地物編集ロックの解除	地物に対する編集ロックを解除して他ユーザによる編集を許可する	Transaction
		地物追加	新しい地物を登録する	(Transaction)
		地物削除	既存の地物を削除する	(Transaction)
		地物編集	既存の地物を編集する	(Transaction)
	表示	レイヤの描画設定の登録	描画スタイル、表示順序などの描画設定を登録する	(GetMap、GetMapUrl)
		レイヤの表示状態登録	レイヤの表示／非表示の状態を登録する	(GetMap、GetMapUrl)
		縮尺の取得	地図画像の縮尺を取得する	GetScale
		強調表示地物の設定	強調表示する地物を設定する	(GetMap、GetMapUrl)
		表示時点設定	地図画像を作成する時点を設定する	(GetMap、GetMapUrl)
		地図画像取得	座標範囲、画像サイズ、画像フォーマット、縮尺、回転角度を指定して地図画像を取得する	GetMap
		地図画像 URL 取得	座標範囲、画像サイズ、画像フォーマット、縮尺、回転角度を指定して地図画像取得のための URL を取得する	GetMapUrl

サービス分類	機能分類	機能	内容	SOAP 1.1 セッション管理なし	
		索引図画像取得	座標範囲、画像サイズ、画像フォーマット、縮尺、回転角度を指定して索引図画像を取得する	(GetMap、GetMapUrl)	
		地図印刷画像取得	座標範囲、用紙サイズ、画像フォーマット、縮尺、回転角度を指定して印刷用地図画像を取得する	(GetMap、GetMapUrl)	
		凡例画像取得	レイヤの凡例画像を取得する	GetLegendImage	
		ピクセル座標緯度経度座標変換	地図画像上のピクセル単位の座標を緯度経度座標に変換する	ConvertImageCoordinates	
		座標系変換	変換する座標と座標系を指定して座標変換を行う	ConvertCoordinates	
	計測	距離計測	座標列あるいは地物 ID リストを指定して、総延長を取得する	GetLength	
		面積計測	座標列あるいは地物 ID リストを指定して、総面積を取得する	GetArea	
	集計	属性の集計	条件を指定して集計した結果を取得する	TotalFeature	
	データベース	データベース管理	データベース定義情報の取得	データベースの定義情報を取得する	GetLinkDBMetadata
			データの取得	データベースに対して条件を指定してデータを取得する	GetLinkDBData
データの直接取得			データベースに対して SQL 文を実行した結果を取得する	GetLinkDBSQL	
()は、別の機能で定義されている関数であることを示す					

1-2-1-2. SOAPとWSDLのための関数仕様(セッション管理あり)

以下に、セッション管理ありの場合の SOAP による地理情報のインタフェースの関数仕様の一覧を示す。

附-表 1-3 SOAP による地理情報のインタフェース (セッション管理あり)

サービス分類	機能分類	機能	内容	SOAP 1.1 セッション管理あり
GIS	初期設定	サービスメタデータ取得	利用可能なサービスの情報を取得する	GetCapabilities
		ログイン	セッションを確立してサービスの利用を開始する	Initialize
		ログアウト	セッションを破棄してサービスの利用を終了する	Terminate
		データ範囲取得	全データの座標範囲を取得する	GetExtent
		地物定義情報取得	地物の定義情報を取得する	GetFeatureClassInfo
	検索	ID 指定の地物検索	地物の ID を指定して検索し、発見された地物の情報を取得する	GetFeature
		点座標指定の地物検索	点の座標を指定して検索し、発見された地物の情報を取得する	(GetFeature)
		面指定の地物検索	面の座標を指定して検索し、発見された地物の情報を取得する	(GetFeature)
		属性指定の地物検索	地物の属性に条件を設定して検索し、発見された地物の情報を取得する	(GetFeature)
		属性コードリスト取得	コードリスト型の属性項目が取りうる値のリストを取得する	GetCodeList
	編集	地物編集ロック	地物に対して編集ロックをかけて他ユーザによる編集を禁止する	LockFeature
		地物編集ロックの解除	地物に対する編集ロックを解除して他ユーザによる編集を許可する	Transaction
		地物追加	新しい地物を登録する	(Transaction)
		地物削除	既存の地物を削除する	(Transaction)
		地物編集	既存の地物を編集する	(Transaction)
	表示	レイヤの描画設定の取得	レイヤの描画スタイルを取得する	GetLayerStyle
		レイヤの描画設定の登録	レイヤの描画スタイルを登録する	SetLayerStyle
		レイヤの表示状態の取得	レイヤの表示／非表示の状態を取得する	GetLayerState
		レイヤの表示状態登録	レイヤの表示／非表示の状態を登録する	SetLayerState
		縮尺の取得	地図画像の縮尺を取得する	GetScale
		現在の縮尺の取得	直前に表示した地図画像の縮尺を取得する	(GetScale GetMapUrl)
		強調表示地物の設定	強調表示する地物を設定する	SetHighlightFeature

サービス分類	機能分類	機能	内容	SOAP 1.1 セッション管理あり	
		表示時点設定	地図画像を作成する時点を設定する	(GetMap GetMapUrl)	
		地図画像取得	座標範囲、画像サイズ、画像フォーマット、縮尺、回転角度を指定して地図画像を取得する	GetMap	
		地図画像 URL 取得	座標範囲、画像サイズ、画像フォーマット、縮尺、回転角度を指定して地図画像取得のための URL を取得する	GetMapUrl	
		索引図画像取得	座標範囲、画像サイズ、画像フォーマット、縮尺、回転角度を指定して索引図画像を取得する	(GetMap GetMapUrl)	
		地図印刷画像取得	座標範囲、用紙サイズ、画像フォーマット、縮尺、回転角度、タイトル、方位、スケールバーを指定して印刷用地図画像を取得する	(GetMap GetMapUrl)	
		凡例画像取得	レイヤの凡例画像を取得する	GetLegendImage	
		ピクセル座標緯度経度座標変換	地図画像上のピクセル単位の座標を緯度経度座標に変換する	ConvertImageCoordinates	
		座標系変換	変換する座標と座標系を指定して座標変換を行う	ConvertCoordinates	
	計測	距離計測	座標列あるいは地物 ID リストを指定して、総延長を取得する	GetLength	
		面積計測	座標列あるいは地物 ID リストを指定して、総面積を取得する	GetArea	
		集計	属性を指定して集計した結果を取得する	TotalFeature	
	データベース	データベース管理	データベース定義情報の取得	データベースの定義情報を取得する	GetLinkDBMetadata
			データの取得	データベースに対して条件を指定してデータを取得する	GetLinkDBData
データの直接取得			データベースに対して SQL 文を実行した結果を取得する	GetLinkDBSQL	
()は、別の機能で定義されている関数であることを示す					

1-2-2. 詳細情報の入手先

「河川 GIS・河川アプリケーション標準インタフェースガイドライン」は、(財)河川情報センターの Web サイト(河川 GIS・河川アプリケーション標準インタフェース作成協議会のページ)上で公開されている。上記で紹介した SOAP による地理情報のインタフェースの実装仕様等の詳細に関しては、下記 URL よりダウンロードできる同ガイドライン本文を参照されたい。

http://www.river.or.jp/setumei/result_gis.html (平成 19 年 8 月時点)

2. ガイドラインの適用事例

本ガイドラインの適用事例として、次項より 2-1 システム実装の事例及び 2-3 インタフェース策定の事例の 2 種類の適用事例を示す。

システム実装の事例では、本ガイドラインで定めたインタフェース(関数仕様及び実装仕様)に準じてシステムを実装する場合の具体例として、「建設情報ポータルサイト」の事例を示す。

インタフェース策定の事例では、本ガイドラインを「汎用的な標準インタフェースのガイドライン」と位置づけ、河川という応用分野における関数仕様の拡張定義事例、電子納品保管管理システムという具体的なシステムのインタフェース仕様の関数仕様定義事例を示す。

2-1. システム実装の事例

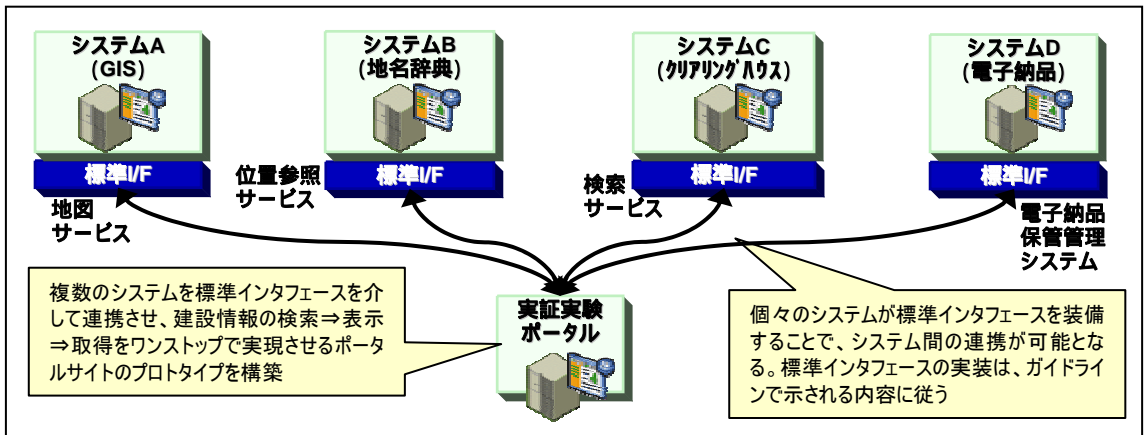
本ガイドラインの適用事例や具体的な利用方法の事例として、実証実験（建設情報ポータル）の例を示す。

2-1-1. 概要

実証実験システム（建設情報ポータルサイト）では、4つの既存システムを連携させた。連携させた既存システムには、全て標準インタフェースが装備されており、建設情報ポータルは、4つの既存システムが持つ機能を標準インタフェースを介して利用し、それぞれのシステムの一部の機能を組み合わせて建設情報ポータルを構成している。

附-表 2-4 連携システムの概要

No.	システム	標準インタフェース	備考
A	電子国土Webシステム (GIS)	ISO19128 (WMS : Web Map Server Interface)	国土地理院の研究成果を借用
B	地名辞典(位置参照)	JIS X 7112 の拡張版(「時空間情報利活用システム推進事業」において検討された PI(Place Identifier)の運用モデルで定義されたインタフェース)	財)日本情報処理開発協会データベース振興センターより研究成果を借用
C	クリアリングハウス(メタデータ検索)	OGC Catalogue Service (CSW : Catalogue Service for Web)	国土地理院の研究成果を借用
D	電子納品保管管理システム(建設情報の取得)	電子納品保管管理システムが有するインタフェース	附-表 2-8 参照
	実証実験システム(建設情報ポータルアプリケーション)	自らはインタフェースを持たないが、A～D のシステムが有する標準インタフェースを利用し、各システムを連携させたアプリケーションとして構築。	



附-図 2-1 実証実験システムの概要

2-2. 連携システムの詳細

2-2-1-1. システム A (GIS)

システム A は、GIS (地理情報サービス) である。今回の実証実験システムの構築にあたっては、電子国土 Web システムを利用した。

通常の電子国土 Web システムは、Web ブラウザと電子国土プラグインを利用し、提供・公開されている「電子国土 API」を利用して地図を表示するアプリケーションを構築できる。しかし、電子国土 API は、電子国土プラグインを操作するための API であり、標準インタフェースではないため、他のアプリケーションからの利用は容易ではない。

電子国土 Web システムを公開・配布している国土地理院では、実用化はしていないものの、ISO19128 (WMS) に基づくインタフェースを装備した電子国土 Web システムを開発済みである (国土地理院技術資料等「電子国土 WMS マネージャ及びグラフィックプロセッサ」。以下、WMS 版電子国土)。今回の実証実験システムの構築にあたっては、国土地理院より WMS 版電子国土を借用し、これを地理情報サービスとして利用した。

以下に、WMS 版電子国土の持つ標準インタフェースの概要を示す。

附-表 2-5 WMS 版電子国土の標準インタフェース

関数の名称	関数の概要
GetCapabilities	サービス自身の概要を記したサービスメタデータを返す
GetMap	リクエスト内容に応じた地理情報を、デジタル画像として返す

2-2-1-2. システム B(地名辞典)

システム B は、地名辞典(位置参照サービス)である。今回の実証実験システムの構築にあたっては、「時空間情報利活用システム推進事業」において財団法人日本情報処理開発協会データベース振興センターが開発した実証実験システムの一部を借用し、これを利用した。

以下に、地名辞典が持つ標準インタフェースの概要を示す。

附-表 2-6 地名辞典の標準インタフェース

関数の名称	関数の概要
GetCapabilities	サービス自身の概要を記したサービスメタデータを返す
SearchLRS	登録されている地理識別子による空間参照系を検索する
GetLRS	登録されている地理識別子による空間参照系を取得する
SearchLocationType	登録されている場所型を検索する
GetLocationType	登録されている場所型を取得する
GetLocationInstance	登録されている場所インスタンスを取得する
SearchLocationInstanceAssociation	複数場所インスタンス関連を検索する

2-2-1-3. システムC(クリアリングハウス)

システム C は、国土地理院が運営する「地理情報クリアリングハウス」(メタデータ検索サービス)である。実用化はされていないが、国土地理院が「JIS X 0806を使用したクリアリングハウスに関する調査・研究作業(国土地理院技術資料 E-1-No.304)」において開発した標準インタフェース(OGC Catalogue Serviceに基づく標準インタフェース)を借用し、これを利用した。

以下に、クリアリングハウスが持つ標準インタフェースの概要を示す。

附-表 2-7 クリアリングハウスの標準インタフェース

関数の名称	関数の概要
GetCapabilities	サービス自身の概要を記したサービスメタデータを返す
GetRecords	メタデータで使用される型定義を返す
GetRecordById	メタデータの属性として有効な値に関する情報を返す

2-2-1-4. システムD(電子納品保管管理システム)

システム D は、国土交通省が無償配布している「電子納品保管管理システム」である。電子納品保管管理システムでは、他のシステムとの連携を実現するためのインタフェースを用意しているため、今回はそれを利用した。以下にその概要を示す。

附-表 2-8 電子納品保管管理システムの標準インタフェース

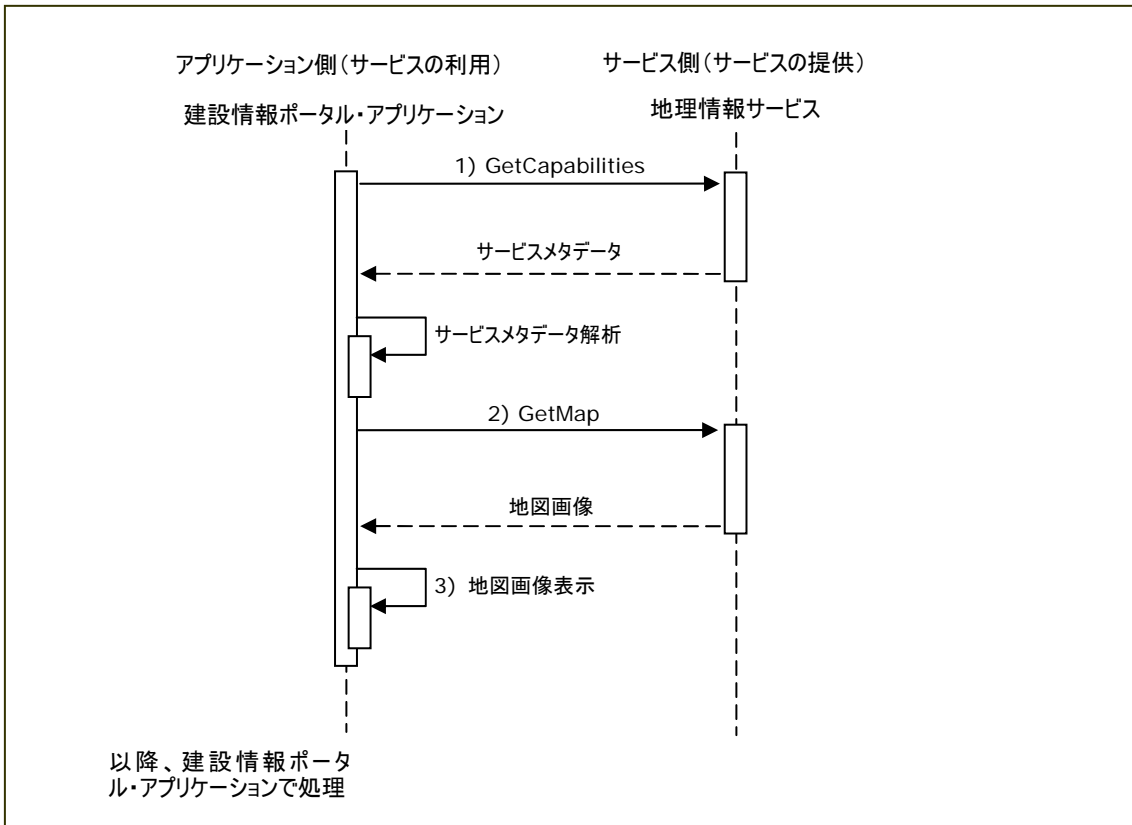
関数の名称	関数の概要
GetDeliveryData	リンクキーを基に電子成果品を検索して一覧表示する。
requestIndex	業務管理ファイル・工事管理ファイルを取得
requestXml	管理ファイルを取得
requestData	データファイルを取得
requestFolder	電子納品フォルダ内のファイルを取得

2-2-2. 実装の手順(例)

以下に、実証実験システム(建設情報ポータル・アプリケーション)が実現する機能と、その機能を構築するために実証実験システムが利用した、各サービスの標準インタフェースの概要を示す。

2-2-2-1. 地図を表示する

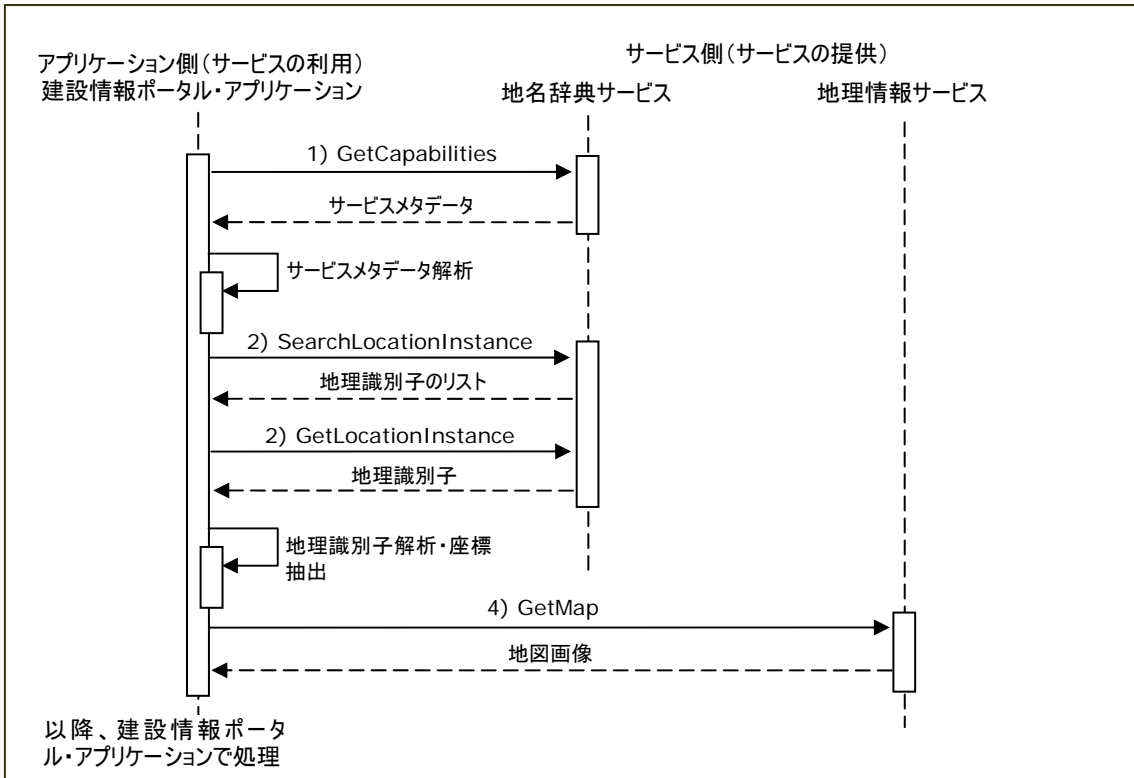
- 1) 地理情報サービス対し、地理情報サービスが実現可能な機能と機能の利用方法を問い合わせる
 - 地理情報サービスに対し、「GetCapabilities」をリクエストすることで、地理情報サービスのサービスメタデータを得る
- 2) 地理情報サービスに対し、表示させたい地図に必要な各種パラメータをリクエストする
 - 地理情報サービスに対し、「GetMap」をリクエストする。リクエストする際は、地図の取得に必要なパラメータを同時に送信する。
 - 表示する地図の範囲、縮尺
 - 表示させるレイヤや地物の種類
 - 凡例の有無
 - 取得する地図画像のファイル形式 など
- 3) 地理情報サービスから、地図画像を得る
 - 2)においてリクエストした内容に基づく地図画像を取得する。取得した地図画像は、建設情報ポータル・アプリケーション上で表示する



附-図 2-2 地図を表示する(シーケンス図)

2-2-2-2. 地名辞典を検索し、検索結果の場所を地図上に表示する

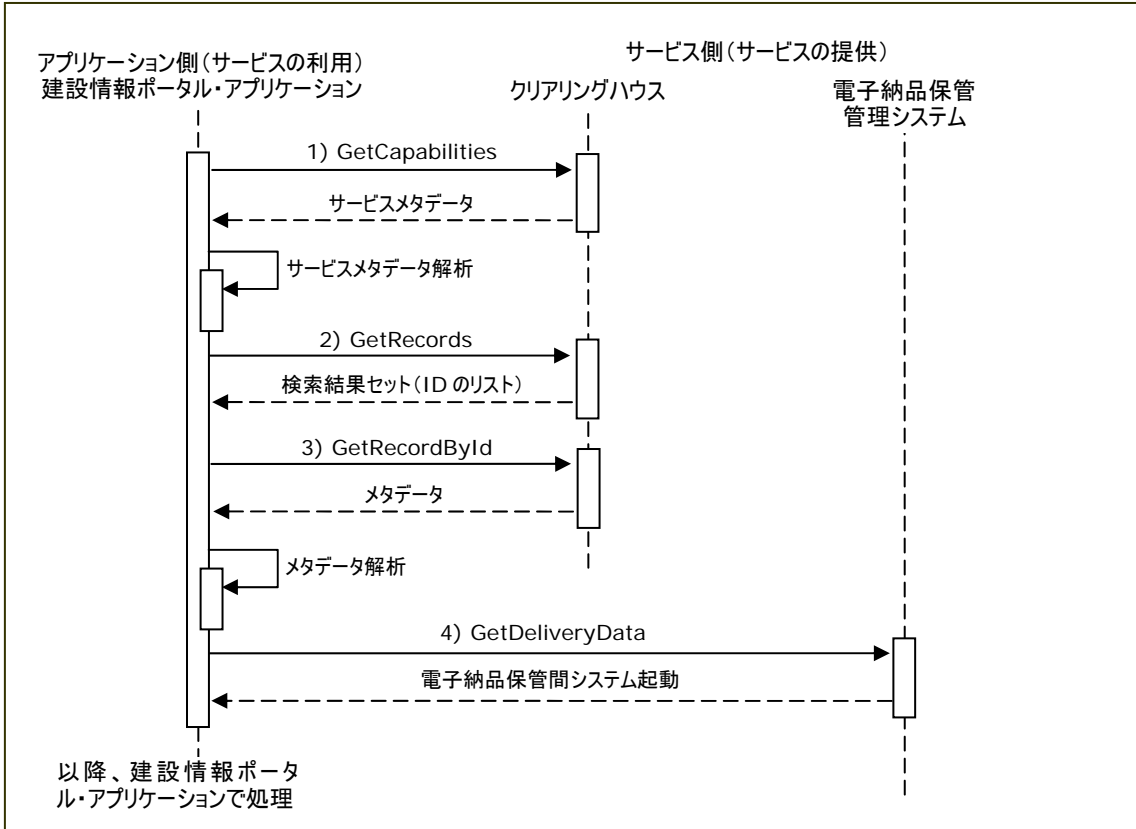
- 1) 地名辞典サービスに対し、地名辞典サービスが実現可能な機能と機能の利用方法を問い合わせる
 - 地名辞典サービスに対し、「GetCapabilities」をリクエストすることで、地名辞典サービスのサービスメタデータを得る
- 2) 地名辞典サービスに対し、検索キーワードで地名検索を実行し、座標を得る
 - 地名辞典サービスに対し、「SearchLocationInstance」をリクエストし、地理識別子の候補一覧を得る。検索のパターンや場合に応じて、「SearchLocationInstance」リクエストを実行する際には、事前に「SearchLocationType」や「GetLocationType」を実行し、場所型を取得しておく
 - 「SearchLocationInstance」の結果を基に、「GetLocationInstance」をリクエストし、地理識別子を取得する
- 3) 座標を基に、該当位置の地図を表示する
 - 「GetLocationInstance」によって取得した地理識別子の情報から、座標値を抽出する
 - 抽出した座標値を基に、地理情報サービスに対して「GetMap」をリクエストし、当該箇所の地図画像を得る
 - 取得した地図画像は、建設情報ポータル・アプリケーション上で表示する



附-図 2-3 地名辞典を検索し、検索結果の場所を地図上に表示する (シーケンス図)

2-2-2-3. メタデータを検索して電子納品成果を表示する

- 1) メタデータ検索サービスに対し、メタデータ検索サービスが実現可能な機能と機能の利用方法を問い合わせる
 - メタデータ検索サービスに対し、「GetCapabilities」をリクエストすることで、メタデータ検索サービスのサービスメタデータを得る
- 2) メタデータ検索サービスに対し、検索キーワードでメタデータ検索を実行し、メタデータを得る
 - メタデータ検索サービスに対し、「GetRecords」をリクエストし、メタデータの候補一覧を得る
 - 取得した候補一覧には、各々のメタデータの ID が記述されている。この ID を、「GetRecordById」とともにメタデータ検索サービスにリクエストすることで、メタデータを得る
- 3) メタデータに記載されている情報から、電子納品保管管理システムを呼び出す
 - メタデータには、電子納品保管管理システムを呼び出すためにリンクキーが記述されている
 - メタデータを取得した建設情報ポータル・アプリケーションは、メタデータの内容を解析し、リンクキーを抽出する
 - 抽出したリンクキーを、「GetDeliveryData」を利用して電子納品保管管理システムにリクエストする
- 4) 電子納品保管管理システムが呼び出され、該当する電子納品成果を得る
 - GetDeliveryData により、リンクキーに該当する情報を表示した状態で、電子納品保管管理システムが呼び出される
 - 電子納品保管管理システムの機能を利用し、電子納品成果を確認、取得する



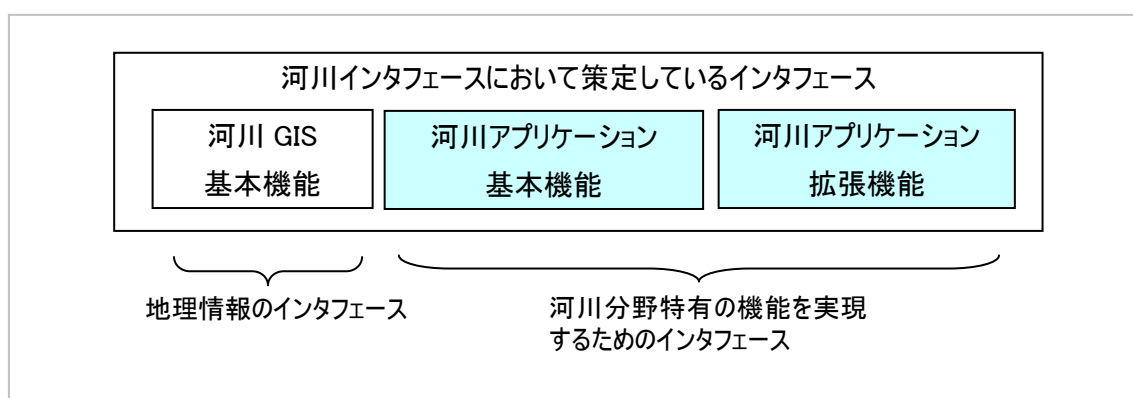
附-図 2-4 メタデータを検索して電子納品成果を表示する (シーケンス図)

2-3. インタフェース策定の事例

2-3-1. 河川 GIS・河川アプリケーション標準インタフェースガイドラインにおける事例

2-3-1-1. 概要

河川 GIS・河川アプリケーション標準インタフェースガイドライン(以下、河川インタフェース)では、本ガイドラインで定める地理情報のインタフェースのほか、河川分野での利用を考慮したインタフェースを定義している。



附-図 2-5 河川インタフェースにおける 3 種類のインタフェース

2-3-1-2. 河川インタフェースで策定している関数

河川分野特有のインタフェースとして、以下の関数を定義している。

附-表 2-9 河川インタフェースにおいて定義している河川分野特有のインタフェース

サービス分類	機能分類	機能	内容	SOAP 1.1
観測データ	データ取得	雨量データ取得	雨量データを取得する	GetRainfallData
		河川水位・流量データ取得	河川水位・流量データを取得する	GetFlowData
解析	解析	浸水予測結果取得	指定した範囲の浸水予測情報(浸水深)を取得する	GetInundationHazardousData
		流下・流入範囲取得	地点を指定し、その流下範囲または流入範囲を取得する	GetFlowArea
	地形処理	標高データ取得	標高データを取得する	GetHeightData
		鳥瞰図取得	三次元処理された鳥瞰画像を取得する	GetView
		任意地形断面取得	断面位置を指定して、地形断面を取得する	GetProfile

2-3-1-3. 詳細情報の入手先

「河川 GIS・河川アプリケーション標準インタフェースガイドライン」は、(財)河川情報センターの Web サイト(河川 GIS・河川アプリケーション標準インタフェース作成協議会のページ)上で公開されている。上記で紹介した事例の詳細に関しては、下記 URL よりダウンロードできる同ガイドライン本文を参照されたい。

http://www.river.or.jp/setumei/result_gis.html (平成 19 年 8 月時点)

2-3-2. 電子納品保管管理システムにおける事例

2-3-2-1. インタフェースの概要

電子納品保管管理システムにおけるインタフェースは、元来電子納品保管管理システム用に定義されたものであるため、標準インタフェースそのものではないが、関数仕様の策定方法等に関しては本ガイドラインとほぼ同様のルールで作成されている。

2-3-2-2. 関数の一覧

以下に、電子納品保管管理システムが装備しているインタフェースの関数一覧を示す。これらの関数は、標準インタフェースそのものではないが、本ガイドラインにおいて提示している関数仕様のありかたとほぼ同様の考え方にに基づき策定されたものである。

附-表 2-10 電子納品保管管理システムの標準インタフェース

種別	機能の概要	関数名
B/C	リンクキーを基に電子成果品を検索して一覧表示	GetDeliveryData
B/C	業務管理ファイル・工事管理ファイルを取得	requestIndex
B/C	管理ファイルを取得	requestXml
B/C	データファイルを取得	requestData
B/C	電子納品フォルダ内のファイルを取得	requestFolder
【種別凡例】		
A: 提供するサービスの内容を取得するための関数		
B: 情報を検索するための関数		
C: 情報を取得するための関数		
D: 情報を登録するための関数		

2-3-2-3. 詳細情報の入手先

電子納品保管管理システムは国土交通省が無償で配布している。上記で紹介した事例の詳細に関しては、下記 URL より入手可能な電子納品保管管理システムの配布物一式に含まれる同システムの設計書を参照されたい。

<http://www.mlit.go.jp/tec/it/cals/arcsys/index.htm>（平成 19 年 8 月時点）

地名辞典の整備・運用ガイドライン(案)
第1版

平成19年10月

国土交通省

はじめに

建設情報の多くは位置情報と関連づけることができるため、建設情報の検索には位置情報を連携キーとするのが得策である。コンピュータが位置を特定する最も有効な情報は緯度・経度であるが、人が位置を検索するキーとしては利用しにくい。住所や地名、施設名等、普段に位置を識別する情報（地名）をキーとして、建設情報を検索することが一般的である。

地名辞典とは、住所や地名などの“地理識別子”と経緯度情報などの“座標”とを関連づけた語彙集であり、位置情報に関する辞典である。住所から緯度・経度への変換、郵便番号から住所への変換といった位置情報の変換は、すでに GIS などで実現されているが、変換の方法などについて標準的な共通の決まりは存在しない。地名辞典を整備・更新する際に必要となる考え方及び守るべきデータ構造など、標準的な共通の決まりに従って作成された地名辞典は重要な地理空間情報基盤と言える。地名辞典の整備により、人間が理解しやすい住所や施設の名称を、コンピュータが読み込みやすい座標値に置き換えることで、GIS はもとより、さまざまなアプリケーションでの位置情報の利活用性が向上する。

今後地名辞典はさまざまな機関で作成されることが予想されるが、データ構造や整備・更新する際に守るべき運用ルールなどを標準化することで、異なるシステムでの地名辞典の利用や、重複のない効率的な地名辞典の整備が可能となる。

本書は、地名辞典の整備・運用の考え方、データ構造、運用ルールを規定した資料であり、地名辞典を整備、作成する方、地名辞典を利用する方及び地名辞典を利用したサービス・アプリケーションを開発する方を対象として作成した。本書を利用して標準的な共通の決まりに従った地名辞典が整備され、これがポータルサイト等で広く公開されることで、分散管理されている建設情報の検索、利用が進んでいくこと期待する。

本書の策定にあたっては、国土交通省が（財）日本建設情報総合センターに業務を委託し、建設情報標準化委員会 電子地図／建設情報連携小委員会（柴崎亮介小委員長）に、ご検討・ご審議をいただいた。多大なご協力をいただいた関係各位に深く謝意を表する次第である。

平成 19 年 10 月

国土交通省

<目次>

1. 地名辞典の整備・運用ガイドラインの概要	1
1-1. 目的	1
1-2. 適用範囲	1
1-3. 本ガイドラインの利用方法	2
1-4. 引用規格	4
1-5. 用語の定義	4
2. 地名辞典の整備・運用の考え方	6
2-1. 建設情報の連携における地名辞典の位置づけ	6
2-2. 地名辞典の標準化の重要性	9
2-3. PI (Place Identifier) 仕様の概要	9
2-4. 本ガイドラインとPI 仕様との関係	12
2-5. 地名辞典の整備・運用	13
3. 地名辞典等の整備・運用のための規約	15
3-1. 地名辞典等が守るべきデータの構造	15
3-2. データ作成時・運用時に守るべき規約	25
4. 地理識別子への位置情報の与え方	28
4-1. 地理識別子に付与する位置情報の形状	28
4-2. 位置情報を付与する際の座標系及び座標値の記述	31
5. 地名辞典管理サービスの実装	33
5-1. 地名辞典管理サービスの機能の概要	33
5-2. 地名辞典管理サービスの運用規約	34
附属書「建設分野における地名辞典の整備・運用方法」	附属書- 1
1. 建設分野における地名辞典の整備・運用の流れ	附属書- 1
2. 地名辞典の初期整備方法	附属書- 3
2-1. 整備する地理識別子の選定	附属書- 3
2-2. 地理識別子による空間参照系及び場所型の作成	附属書- 16
2-3. 地理識別子の作成	附属書- 66
2-4. 地名辞典管理サービスへの地名辞典の登録	附属書- 103
3. 地名辞典の運用方法	附属書- 104
3-1. 地名辞典の更新	附属書- 104
4. 地名辞典の拡充	附属書- 113

1. ガイドラインの概要

1-1. 目的

近年様々な建設情報が各機関でデータベース化されているが、分野を跨いだ連携は必ずしも十分に行われていないのが現状である。

一般的に、建設情報は経緯度などの“地理座標”や住所・施設名などといった“地理識別子”等の位置と関連するものが多く、地理座標と地理識別子とが対になった情報をデータベースとして整備すれば（これを地名辞典という）、異なるシステム間でも場所を同じくする複数の情報を一括で検索することが可能となり、異なるシステムで管理されている様々な建設情報を結びつけ、効率的に建設情報を探ることが出来るようになる。ただし、このような地名辞典が異なるシステム間で利用されるためには、地名辞典が標準的な共通の決まりにしたがって各機関で作成される必要がある。

「地名辞典の整備・運用ガイドライン」(以下、本ガイドラインと呼ぶ)は、建設情報を結びつけ、建設情報を効率的に探すために重要となる地名辞典を整備・運用する際に必要となる考え方及び標準的な共通の決まりを示す。

1-2. 適用範囲

本ガイドラインは、標準的な共通の決まりにしたがって地名辞典を整備・更新するために必要な、以下の点について示す。

- ✓ 地名辞典を整備・運用する際に必要な考え方
- ✓ 地名辞典が守るべきデータ構造
- ✓ 地名辞典を整備・更新する際に守るべきルール

また、本ガイドラインの附属書では、「建設分野における地名辞典の整備・更新方法」を示しており、建設情報を利用する国及び地方公共団体の職員、測量・調査・設計・施工に携わる民間の企業及び個人が、建設分野における地名辞典を整備する場合及び整備した地名辞典を更新する場合に、本ガイドラインを適用することができる。

1-3. 本ガイドラインの利用方法

図 1-1 に、地名辞典の新規整備に向けた「整備する地理識別子の選定」「地名辞典の整備」から整備後の「地名辞典の運用」「地名辞典の拡充」に至る、地名辞典を整備・運用していくうえでの一連のサイクルを示す。本ガイドラインでは、本編または附属書の中で、整備・運用の各段階において守るべき事項や指針を示すことで、どの段階においても本ガイドラインを利用できるような構成としている。

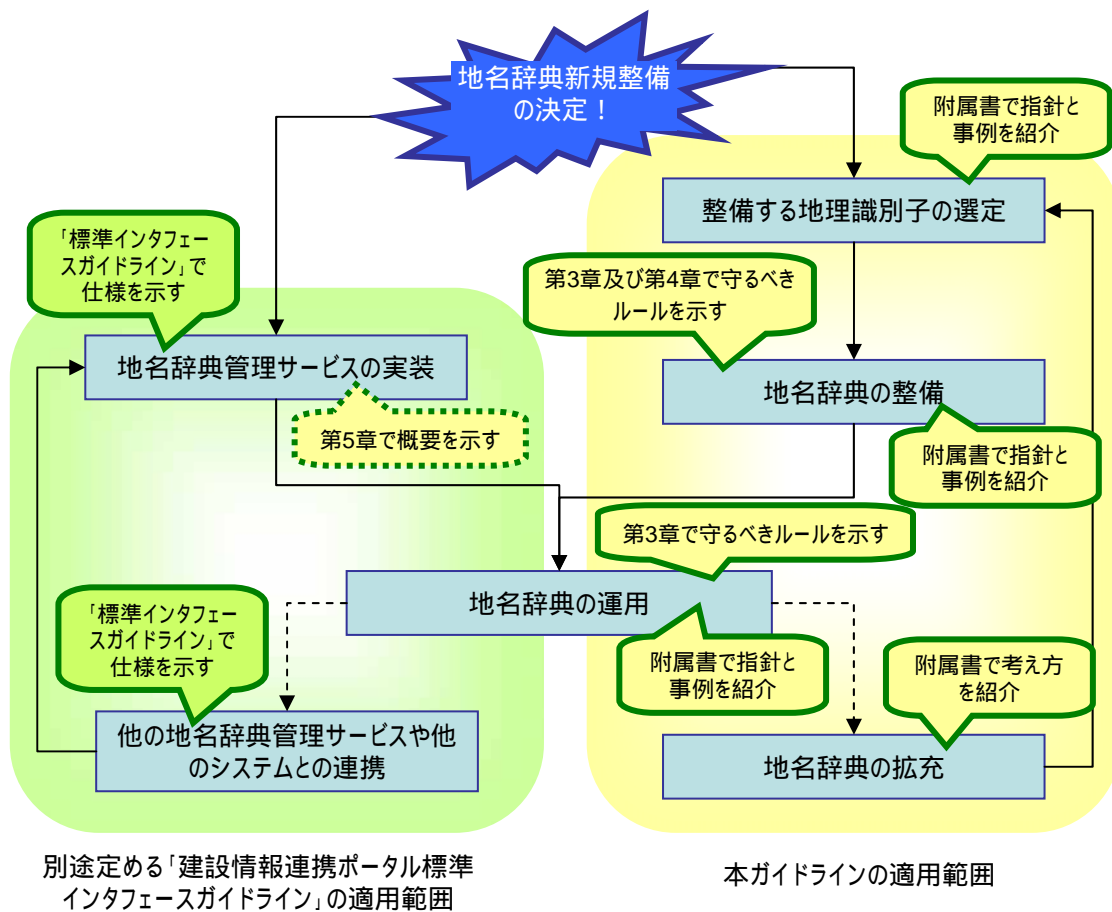


図 1-1 地名辞典の整備・運用の流れ

また、本ガイドラインの利用対象者は、表 1-1 のようになり、利用対象者が本ガイドラインを利用する際に参照するとよい項目を、表 1-2 に示す。

表 1-1 本ガイドラインの利用対象者

利用対象者	対象者の内容
地名辞典の整備主体	地名辞典を作成するひと。
地名辞典の利用者	地名辞典を利用するひと。また、地理識別子の更新を行うひと。
地名辞典を利用したサービス・アプリケーションの開発者	別途定める標準インタフェースを実装して地名辞典サービスを開発するひと。また、地名辞典サービスを利用したアプリケーションを開発するひと。

表 1-2 利用対象者が参照する本ガイドラインの項目

本ガイドラインの目次	整備主体	利用者	開発者
1.地名辞典の整備・運用ガイドラインの概要			
1-1.目的			
1-2.適用範囲			
1-3.本ガイドラインの利用方法			
1-4.引用規格			
1-5.用語の定義			
2.地名辞典の整備・運用の考え方			
2-1.建設情報の連携における地名辞典の位置付け			
2-2.地名辞典の標準化の重要性			
2-3.PI (Place Identifier) 仕様の概要			
2-4.本ガイドラインと PI 仕様との関係			
2-4.地名辞典の整備・運用			
3.地名辞典等の整備・運用のための規約			
3-1.地名辞典等が守るべきデータの構造			
3-2.データ作成時・運用時に守るべき規約			
4.位置情報の与え方			
4-1.地理識別子に付与する位置情報の形状			
4-2.位置情報を付与する際の座標系及び座標値の記述			
5.地名辞典管理サービスの実装			
5-1.地名辞典管理サービスの機能の概要			
5-2.地名辞典管理サービスの運用規約			
附属書 建設分野における地名辞典の整備方法			

1-4. 引用規格

本ガイドラインは、以下に掲げる規格・仕様を引用する。

PI 仕様書

(2007年4月 (財)日本情報処理開発協会データベース振興センター(JIPDEC/DPC))

JIS X 7112 地理情報 - 地理識別子による空間参照

1-5. 用語の定義

本ガイドラインで使用する用語とその定義を以下に示す。

地理識別子

場所の識別のための、ラベルまたはコードの形式による空間参照。(JIS X 7112)

PI (Place Identifier)

場所の識別子。地理識別子や座標、URI、ISBN など、様々な識別子のうち、場所の識別に用いることが可能な識別子。

PIについては、第2章において詳細に説明する。

場所インスタンス

地理識別子と対応する位置情報との対をもったインスタンス。

地名辞典

位置に関する何らかの情報を含む、地物のクラスのインスタンスの要覧。(JIS X 7112)

空間参照系

実世界における位置を記述するための体系。(JIS X 7112)

空間参照系には、座標による空間参照系と地理識別子による空間参照系の2種類がある。

地理識別子による空間参照系

地理識別子によって実世界における位置を記述するための体系。

場所型

場所インスタンスの特性を定めた型。

インタフェース

実体の振る舞いを特徴づける名前の指定された操作の集合。(ISO19119)

サービス

インタフェース経由で実体によって提供される他と明確に区分される機能の一部。
(ISO19119)

地名辞典管理サービス

一つ以上の地名辞典を直接的あるいは間接的に登録・管理し、地名辞典に存在する場所インスタンスに記述された地理識別子を検索・提供、あるいは別の PI に変換する操作を提供するサービス。地名辞典管理サービスは、PI 変換サービスともいう。

空間参照系管理サービス

一つ以上の空間参照系を登録・管理し、場所インスタンスの型である場所型を検索・提供したり、あるいはある座標参照系から別の座標参照系への変換情報を提供するサービス。

2. 地名辞典の整備・運用の考え方

この章では、地名辞典を整備・運用していくにあたって必要な考え方を示す。

まず、建設情報の連携における地名辞典の位置付けや地名辞典の標準化の重要性を述べ、本ガイドラインでデータ構造等を引用する PI (Place Identifier) 仕様の概要や、本ガイドラインと PI 仕様との関係を示す。

最後に、まとめとして地名辞典の整備・運用時において考えるべき事項を示す。

2-1. 建設情報の連携における地名辞典の位置づけ

2-1-1. 地名辞典とは

地名辞典とは、地理識別子とそれに対応する位置情報のセットを複数格納した“辞典”のことをいう。例えば、問い合わせのあった“北柏高架橋”の場所が分からない人が“北柏高架橋”をキーワードに検索をかけると、地名辞典は自身もつ辞典のなかから該当する地理識別子がないかを探し出し、ある場合は、対応する位置情報をもとに例えばその周辺の地図を表示することができる。

このような仕組みは、特に新しいわけではなく、業務支援 GIS やインターネットの地図検索サービス等で、既に住所検索や目標物検索として利用されている技術である。

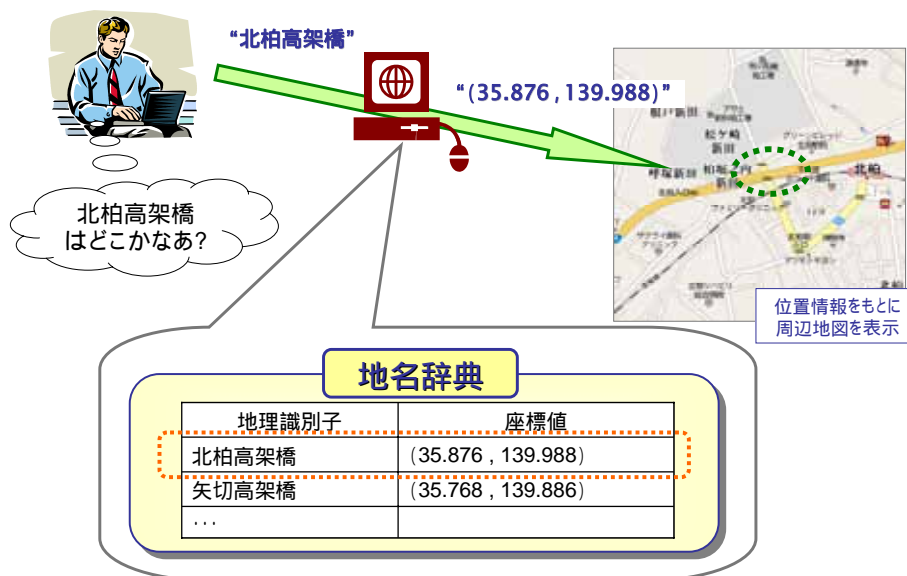


図 2-1 地名辞典の利用イメージ

さらに、整備した地名辞典を利用することで、情報検索をより効率よく、かつより高度に行うことができ、業務効率の面でも大変有効である。

地名辞典を使って、既存の台帳や帳票の利用性の向上！～災害時の利用イメージ～

災害時に連絡のあった「被災箇所一覧表」

被災箇所	位置
1 国道 号線 市 × × 地先	31.5kp
2 国道 × 号線 市 地先	32.3kp
3	31.1kp
4 国道 号線 市 × × 地先	...

日常、台帳として管理している「緊急時支援施設一覧表」

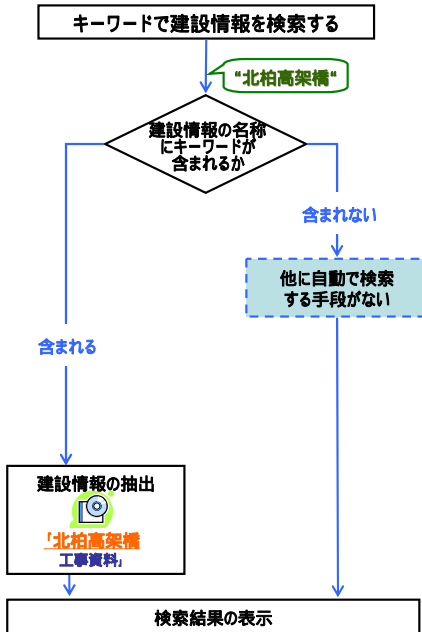
施設名	住所
A 大学病院	柏市 町1-1
B 市民病院	柏市 × × 町3-5
C 大学	柏市 町1-11
D × × 公園	...



地名辞典を使って、欲しい情報をより効率よく収集！

「北柏高架橋」の近くで損壊事故が発生したとの通報を受けた!!
周辺箇所の過去の工事図面や設計資料が欲しい!!!

地名辞典がない場合の建設情報の検索



地名辞典がある場合の建設情報の検索

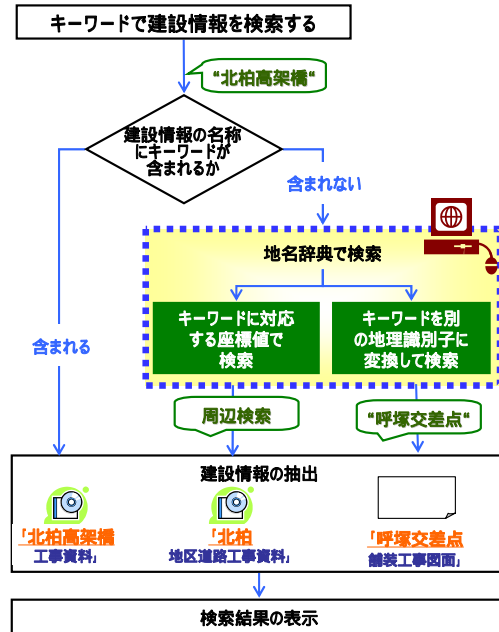


図 2-2 地名辞典の利用による情報検索の向上イメージ

2-1-2. 地名辞典の整備の現状

現状では、地名辞典に相当するものが数多く存在すると考えられるが、それらは独自の仕様でデータを作成し、特定の GIS やサービスでのみ利用可能な状態となっている。この時、データは整備主体がもつ地図データや市販のデータなどをリソースとして地名辞典のデータを整備し、特定の GIS やサービスがもつ独自のインターフェース（“ 関数仕様及びそれを利用するためのプログラム上の手続きを定めた規約 ”）を用いて地名辞典のデータを利用している。

また、地名辞典の整備や検索機能の開発を独自に行わざるを得ないため、整備費用や維持更新費用・作業等の負担が大きいのが現状である。

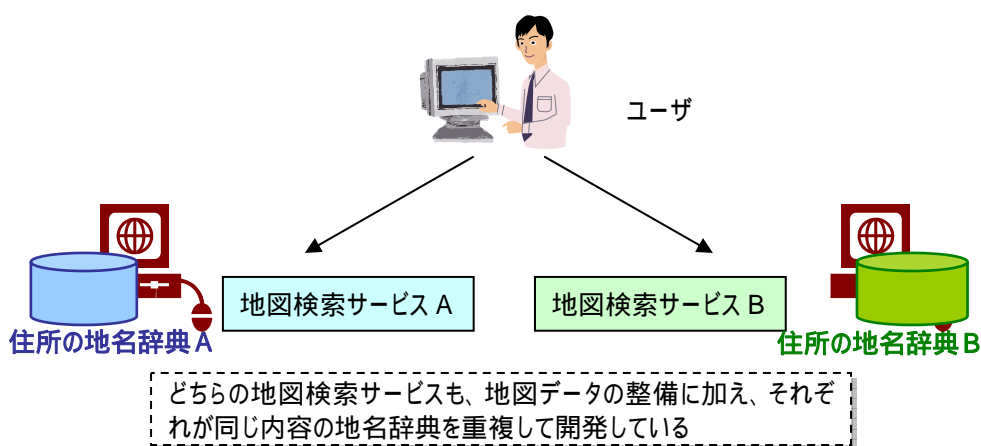


図 2-3 個別の地名辞典整備のイメージ

2-1-3. これからの地名辞典の利用方法

前項のような課題を解決するには、地名辞典やそれを利用したサービスを個別に整備するのではなく、公開・共用していくことが求められる。これにより重複整備の防止や維持・管理負担の軽減による費用削減が実現できるだけでなく、本来のサービス（例：地図検索サービスでは地図の作成）の品質を向上させることが可能である。また、地名辞典を利用したサービスを公開・共用することでユーザを増やすことにより、地名辞典そのものの品質も向上させることができるようになる。

このような地名辞典を利用したサービスの公開・共用を可能とするためには、サービスを提供する場合には標準インターフェースを実装し、共用可能なサービスとして公開することが求められる。

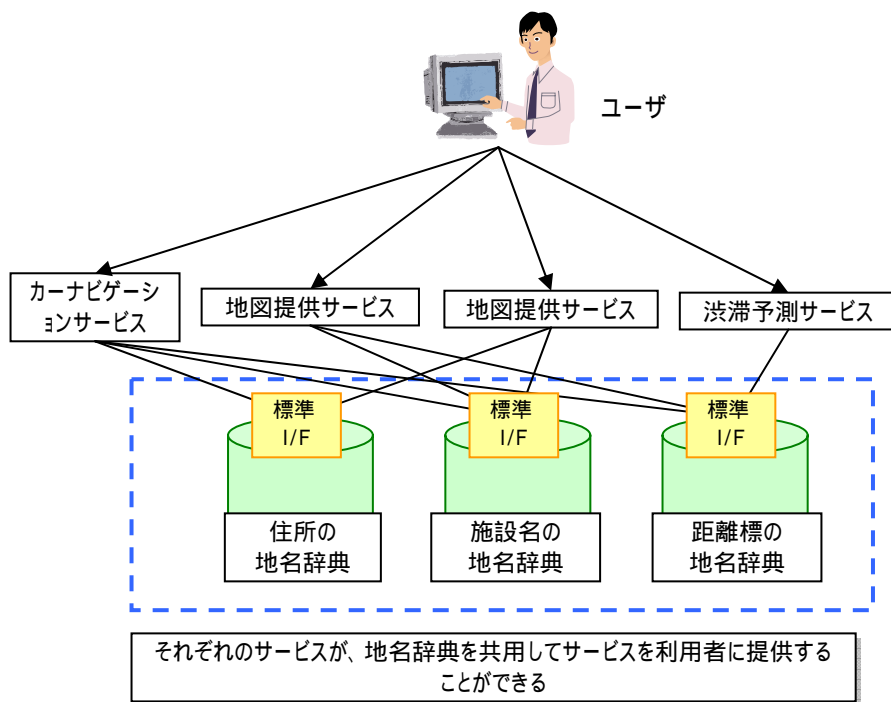


図 2-4 地名辞典を利用したサービスの公開・共用のイメージ

2-2. 地名辞典の標準化の重要性

2-1 で示したように、整備した地名辞典を相互に連携しながら利用できるようにするためには、地名辞典がある共通の標準的な決まりに沿って作成される必要がある。

地名辞典を作成する際の共通の標準的な決まりとして、日本工業規格では、「JIS X 7112 地理識別子による空間参照」(以下、JIS X 7112)を規定し、地名辞典及び地名辞典を整備・運用する上で必要な 3 つのクラス(地理識別子による空間参照系、場所型、場所インスタンス)のデータ構造を規定している。

また、(財)日本情報処理開発協会データベース振興センター(以下、JIPDEC/DPC)では、JIS X 7112 を応用した「PI(Place Identifier)仕様」を検討し、JIS X 7112 に準拠したデータ構造を定めるとともに、PI を WWW (World Wide Web) 空間上で検索、変換して運用するための PI 参照モデルを策定した。

今後整備される地名辞典は、JIS X 7112 に準拠したデータとして整備され、相互に連携しながら利用できるようにすることが非常に重要である。

2-3. PI (Place Identifier) 仕様の概要

PI (Place Identifier) 仕様は、JIPDEC/DPC ((財)日本情報処理開発協会データベース振興センター)が「時空間情報利活用システム推進事業」の中で検討を進めている技術仕様

であり、PI = 場所の識別子を WWW 空間上で登録、検索、変換（同じ場所を示す別の識別子への変換等）し、相互に流通させるための仕組みである。

この技術仕様で検討している仕組み（これを PI 参照モデルという）の概要を、図 2-5 に示す。PI 仕様では、このうち「データ」「サービス」「インタフェース」について仕様を定めている。

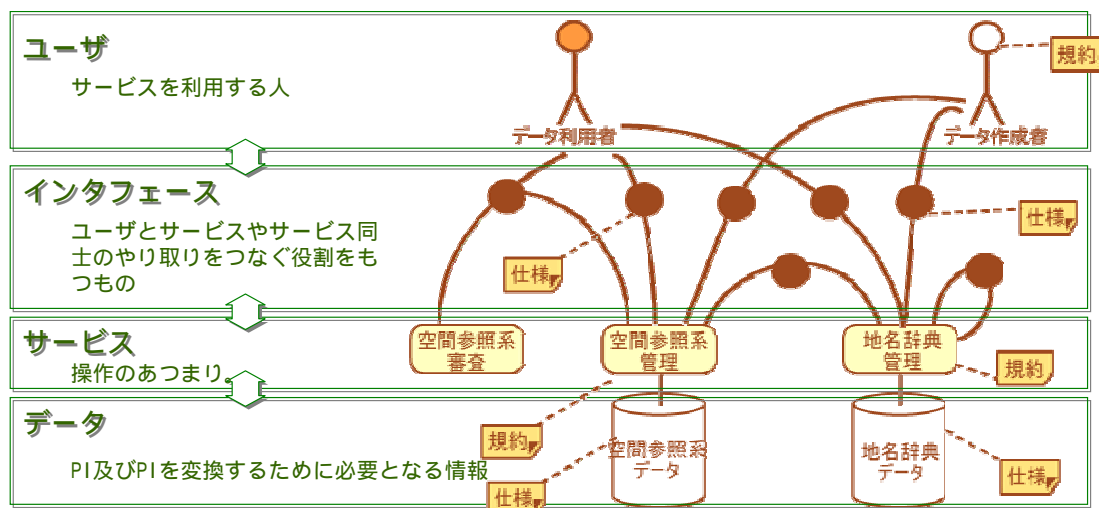


図 2-5 PI 参照モデルの概要

(A) データ

地理識別子を格納する地名辞典や、地理識別子の種類やその内容、記述ルールを示した空間参照系のデータを作成する際の守らなくてはならないデータ構造を定めている。データ構造は、JIS X 7112 に則っている。

また、この仕様で定義している PI は、JIS X 7112 で規定する場所インスタンス及び場所型の仕様を拡張して定義したものである。具体的には、場所インスタンスの geographicIdentifier 属性の型を文字列型から PI に、場所型の identification 属性の型を文字列型から PI_Type に変更することで、PI_Type で定めた符号化規則（記述ルール）に従って明確に地理識別子の値を記述できるようにした。（図 2-6）

なお、データ構造の詳細については、第 3 章で述べる。

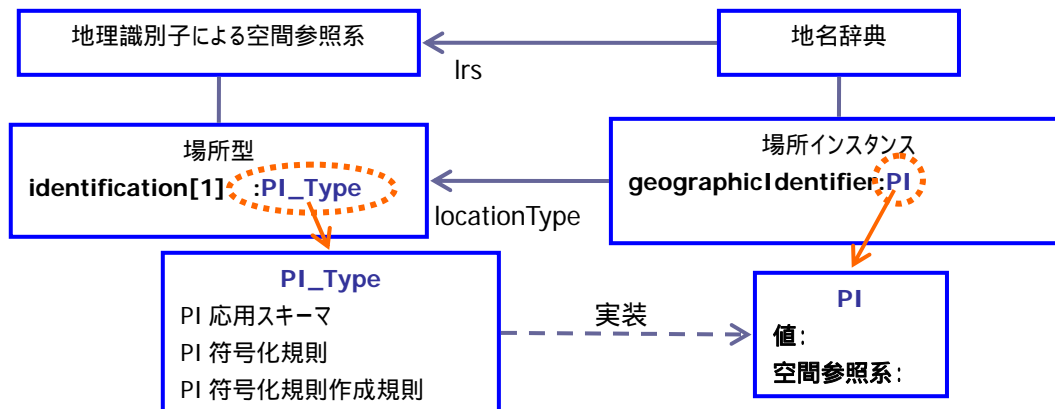


図 2-6 PI のデータ構造の概要

(B) サービス

地名辞典や空間参照系などのデータを管理するとともに、利用者からの問い合わせに対してデータの中から該当する地理識別子を検索し、利用者に提供する機能をもつ。サービスには、地名辞典管理サービス、空間参照系管理サービス、空間参照系登録審査サービスの3種類が存在する。

地名辞典管理サービスは、地理識別子を格納する地名辞典を管理する。そして、定められたインターフェースを実装し、利用者からの問い合わせに対してデータの中から該当する地理識別子を検索し、利用者に提供する機能をもつ。本ガイドラインに沿って作成した地名辞典は、地名辞典管理サービスに登録することで、他のサービスやシステムと連携して利用することが可能となる。地名辞典管理サービスは、複数システム間の情報連携のために、既存のシステムやアプリケーションに追加をするか、あるいは新たにサービスとして作成される必要がある。なお、地名辞典管理サービスは、PI 変換サービスと呼ぶこともある。

空間参照系管理サービスは、地理識別子の種類やその内容、記述ルールを示した空間参照系や場所型を管理する。空間参照系管理サービスも定められたインターフェースを実装し、利用者からの問い合わせに対して空間参照系や場所型の検索を行い、利用者に提供する機能をもつ。空間参照系サービスは複数存在する必要はなく、個々の地名辞典管理サービスが、今後作成される空間参照系サービスを参照すればよい。

空間参照系登録審査サービスは、空間参照系や場所型を新たに作成する際にデータ構造が正しく作成されているかを審査するためのサービスである。ただし、このサービスは求められる要件を検討している段階である（本ガイドライン作成時点）。

PI 参照モデルで定義されるサービスの関係を、図 2-7 に示す。なお、各サービスの詳細については、JIPDEC/DPC で仕様が公開されており、それを参照するとよい。

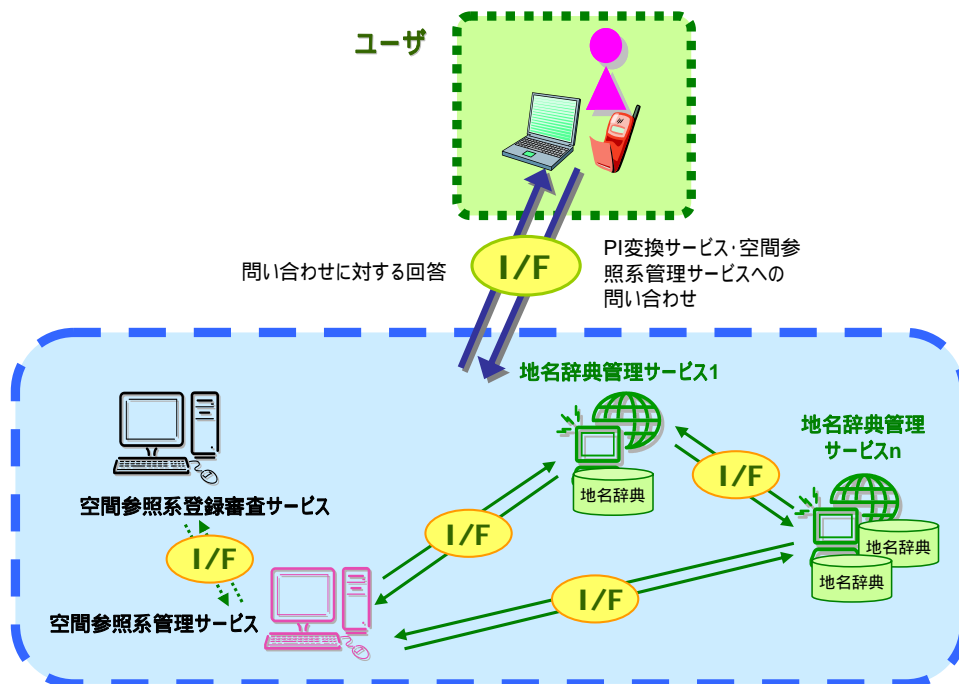


図 2-7 PI 参照モデルで定義される各サービスの関係

(C) インタフェース

アクターとサービス、あるいはサービス同士をつなぐ役割である。サービスに地理識別子や地名辞典を登録する機能、地理識別子を検索する機能を作成する際の仕様（これをインタフェースという）を共通の守るべきルールとし、利用者やサービスの作成者はこの仕様に則ってサービスを実装したり、あるいはサービスにアクセスすることで、他のサービスと簡単に連携が出来るようになる。

インタフェースについても、詳細は JIPDEC/DPC で公開している仕様を参照するとよい。

本ガイドラインでは地名辞典を整備するだけでなく、整備した地名辞典を他のサービスと連携して利用することで建設情報の検索や利用をより効率的に行うことを目指しているため、PI 仕様で示すデータ構造にしたがって地名辞典を作成することとする。

2-4. 本ガイドラインと PI 仕様との関係

本ガイドラインに沿って作成する地名辞典のデータは、PI 仕様のデータ構造に準拠することとするため、本ガイドラインの中でも PI 仕様で定められたデータ構造やデータの運用に関する決まり等に関しては PI 仕様での記述を引用している。

本ガイドラインの中で、PI 仕様から引用している箇所を、表 2-1 に示す。

表 2-1 本ガイドラインにおける PI 仕様からの引用箇所

本ガイドラインの目次		PI 仕様からの引用
1.地名辞典の整備・運用ガイドラインの概要		
1-1.目的		
1-2.適用範囲		
1-3.本ガイドラインの利用方法		
1-4.引用規格		
1-5.用語の定義		
2.地名辞典の整備・運用の考え方		
2-1.建設情報の連携における		
2-2.地名辞典の標準化の重要性		
2-3.PI (Place Identifier) 仕様の概要		
2-4.本ガイドラインと PI 仕様との関係		
2-5.地名辞典の整備・運用		
3.地名辞典等の整備のための規約		
3-1.地名辞典等が守るべきデータの構造		
3-2.データ作成時・運用時に守るべき規約		
4.地理識別子の位置情報の与え方		
4-1.地理識別子に付与する位置情報の形状		
4-2.位置情報を付与する際の座標系及び座標値の記述		
5.地名辞典管理サービスへの地名辞典の登録		
5-1.地名辞典管理サービスの機能の概要		
5-2.地名辞典管理サービスの運用規約		
附属書 建設分野における地名辞典の整備方法		
(…引用、 …一部引用)		

2-5. 地名辞典の整備・運用

2-1～2-4 で示した点を踏まえ、地名辞典の整備・公開・管理の面で考えるべき事項をまとめると以下ようになる。地名辞典を整備・運用する際には以下の点に注意をしていくことが重要である。

地名辞典の整備

- ✓ 地名辞典を整備する際は、JIS X 7112 に準拠したデータとして整備されることが重要である。
- ✓ 本ガイドラインに沿って作成されるデータの構造は、PI 仕様で示すデータ構造にしたがって地名辞典を作成する。具体的なデータ仕様は、第 3 章で示すこととする。
- ✓ 整備にあたっては、データ構造のほかにも守るべきルールがあり、そのルールを守らなければならない。具体的なルールは、第 3 章で示すこととする。

地名辞典の運用

- ✓ 第3章にしたがって整備した地名辞典は、他機関も利用できるように公開するのがよい。
- ✓ 運用にあたっては、別途定める「建設情報連携ポータル標準インタフェースガイドライン」で定められたインタフェースを実装した地名辞典管理サービスに登録する必要がある。(第5章に地名辞典管理サービスの概要を示している)
- ✓ 地名辞典のデータは、GISにおける背景地図データと同様に、データの新鮮さを保つことが重要である。新鮮さを保つためには、整備した地名辞典に対して、施設や路線が新たに建設された場合や名称等が変更になった場合は随時地理識別子の追加や修正を行う必要がある。また、既存リソースの更新時期に合わせて、地名辞典のデータも定期的に更新をすることが必要である。
- ✓ 作成されるデータが多数のシステム間で相互に連携できることを担保するために、地名辞典を管理していく際にも、定められたデータ構造の保持等、守るべきルールがある。
- ✓ 地名辞典の内容を更新する際は、第3章に示すルールを守らなければならない。
- ✓ 地名辞典の内容を更新する際は、地名辞典管理サービスを介して行うのがよい。

3. 地名辞典等の整備・運用のための規約

この章では、地名辞典の整備において必要となる

- ✓ 地名辞典等が守るべきデータの構造
- ✓ 作成・運用時に守るべきルール（以下、規約と呼ぶ）

について示す。

本ガイドラインに沿って地名辞典を整備・運用する場合は、必ずこの章で示す事項を守らなければならない。

3-1. 地名辞典等が守るべきデータの構造

本ガイドラインに沿って作成する地名辞典は、他のサービスと連携し、相互運用させることを考えて、JIS X 7112 に準拠した PI 仕様 に則って作成することとする。

地名辞典のデータ構造の概要を、図 3-1 に示す。図中に青枠で示した 4 つのクラスが、地名辞典作成にあたって必要となるクラスであり、特に、地名辞典（EXPI_Gazetteer）及び場所インスタンス（PI_LocationInstance）は、個々に整備が必要となるクラスである。

図 3-1 で示した 4 つのクラスに加え、地理識別子の値の記述方法と実際の値を記述する PI クラスの 5 つについて、以降にデータ構造の詳細を示す。

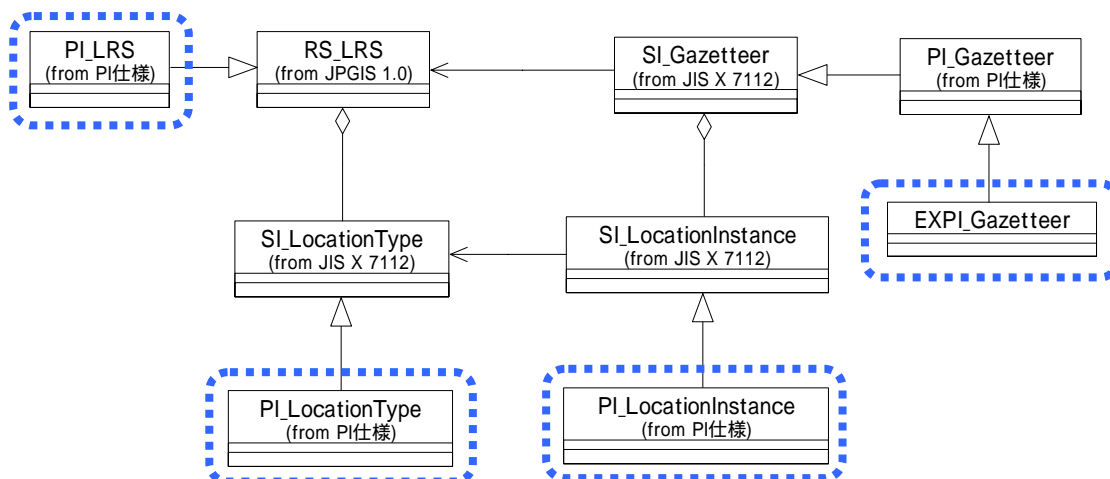


図 3-1 本ガイドラインに沿って作成する、地名辞典のデータ構造の概要

また、地名辞典作成時は、以下の手順で行うこととする。

地理識別子の値をどのように記述するかを「PIタイプ」で定義する。
場所インスタンスに記述する内容・記述方法等をあらかじめ「場所型」で定義する。で定義した「PIタイプ」は、「場所型」の“識別”属性の値として「場所型」に含める。この時、既に定義済みの場所型を利用してもよい。
地理識別子と対応する位置情報の関係を「場所インスタンス」に記述する。これが地名辞典の実データとなる。この時、「場所型」で定義した記述方法に従う。
作成した場所インスタンス(群)を取りまとめて「地名辞典」を作成し、地名辞典の名前や適用範囲、品質等を定義する。
場所型が空間参照を行うための地理識別子であることを、「地理識別子による空間参照系」で定義する。

3-1-1. PI

PIとは、地理識別子の値をより明確に記述するために定められたクラスである。

地理識別子の値は、PIタイプ (PI_Type) のクラスで記述方法を定め、定めた記述方法に従ってPIクラスの“identifier”属性に実際の値を記述する。

図 3-2 に、PIタイプ及びPIのデータ構造を示す。

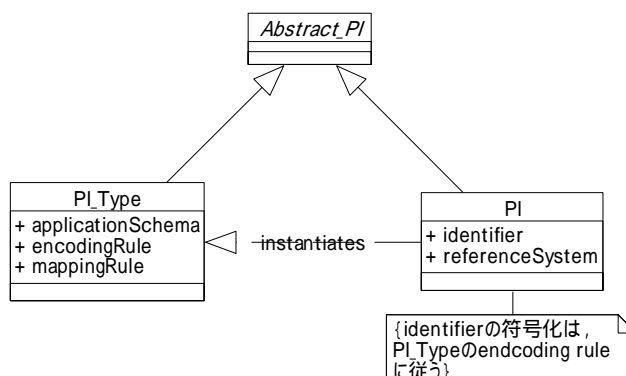


図 3-2 PI のデータ構造

(A) PI 抽象型 (Abstract_PI)

PI 抽象型は、記述方法を定める PI タイプと、そのインスタンスの位置づけとなる PI クラスの上位となるクラスである。抽象型のため、このクラスのデータを作成することはない。

(B) PI タイプ (PI_Type)

PI タイプは、PI の値の記述方法を定めるためのクラスである。PI タイプは、“ 応用スキーマ (applicationSchema)”、“ 符号化規則 (encodingRule)”、“ マッピング規則 (mappingRule)” の 3 つの属性により、PI の値の構造を示すことになる。

PI タイプは、PI_LocationType の “ 識別 (identification)” 属性の型となる。

表 3-1 PI タイプの属性情報

属性 (英語名)	記述条件	記述形式	概要及び定義
応用スキーマ (applicationSchema)	必須	任意 (応用スキーマの識別子あるいは所在が分かる URL 等を記述)	地理識別子の記述方法を定義した応用スキーマ等
符号化規則 (encodingRule)	必須	任意 (符号化規則の識別子あるいは所在が分かる URL 等を記述)	上述の応用スキーマに従って定めた、地理識別子を記述するための符号化規則
マッピング規則 (mappingRule)	必須	任意 (マッピング規則の識別子あるいは所在が分かる URL 等を記述)	上述の符号化規則を定めた根拠となる規則。例えば、「ISO19118」や「JPGIS Ver1.0 附属書 8 で定めた符号化規則」がある。

(C) PI

PI は、地理識別子の値のことである。

PI は PI_LocationInstance の “ 地理識別子 (geogaphicIdentifier)” 属性の型となる。

表 3-2 PI の属性情報

属性 (英語名)	記述条件	記述形式	概要及び定義
識別子 (Identifier)	必須	URI	PI タイプの符号化規則に則って作成された、地理識別子
空間参照系 (referenceSystem)	必須	URI	この地理識別子が属する空間参照系

3-1-2. 地名辞典 (EXPI_Gazetteer)

地名辞典は、地理識別子の実体 (データ) である場所インスタンスの集成として位置付けられる。つまり、複数の地理識別子を取りまとめたデータベースと言い換えることができる。

地名辞典をデータベースとして考えた場合、場所型がデータを格納するテーブル、場所インスタンスがデータベースに格納されるレコードとみなすことができる。したがって、地名辞典クラスの記述内容は、データベースを説明するための情報ということができる。

本ガイドラインで作成する地名辞典は、基本的に PI 仕様で定めたデータ構造に従うが、整備した地名辞典の品質情報を記述できるように PI 仕様で定めたデータ構造を拡張した、“EXPI_Gazetteer” クラスを定義する。

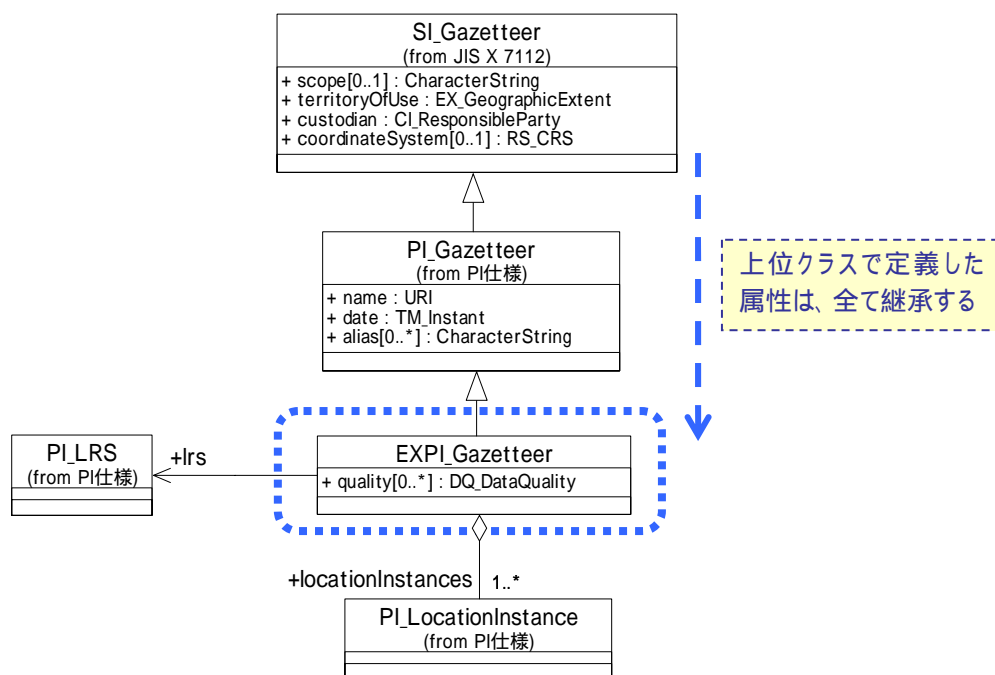


図 3-3 EXPI_Gazetteer の UML クラス図

表 3-3 地名辞典の属性及び関連役割情報

属性及び関連役割 (英語名)	記述条件	記述形式	概要及び定義
名前 (name)	必須	URI	地名辞典の名称を URI を用いて記述
適用範囲 (scope)	任意記述	CharacterString	各地名辞典を適用する範囲を記述する。(使用領域と併せて対象範囲を具体的に説明する)
使用領域 (territoryOfUse)	必須	EX_GeographicExtent	各地名辞典を使用する地域を記述する。(「地理識別子による空間参照系」の有効領域と同様の内容を記述する)
責任者 (custodian)	必須	CI_ResponsibleParty	地名辞典の維持管理に責任をもつ個人名、役職または組織名を記述
座標参照系 (coordinateSystem)	任意記述	RS_CRS	地名辞典が参照する座標参照系を記述してもよい
日付 (date)	任意記述	TM_Instant	地名辞典を作成または更新した日付を記述
別名 (alias)	任意記述	CharacterString	地名辞典の別名を記述
地理識別子による空間参照系 (Irs)	必須	PI_LRS	地名辞典が参照する地理識別子による空間参照系を記述
場所インスタンス (locationInstances)	必須	PI_LocationInstance	地名辞典が格納している場所インスタンスを記述

3-1-2-1. 地名辞典の品質情報

図 3-3 に示す地名辞典のクラスでは、JIS X 7112 で規定されている「名前」、「適用範囲」、「使用領域」、「責任者」、「座標参照系」項目以外に「品質(quality)」を追加している。

「品質」は、地名辞典の利用者が、利用に適したデータであるかを判断するために重要な情報である。地名辞典(地名辞典に格納されている地理識別子)がどのような処理手順で、どのようなデータを元にして作成されたかを説明するため、に示す項目について、表 3-4 の定義に従い、記述する。なお「品質」に関する記述は、拡張規則にもとづいて JMP2.0 の地理情報における品質情報の記述方法に従い、本ガイドラインで定義した。

品質情報そのものは、図 3-4 のとおり任意記述であるが、地名辞典がどのような位置正確度を持ち、どのようなデータをもとに作成されているか等を把握するための重要な情報であるため、可能な限り詳細に記述することが望ましい。

表 3-4 品質情報の記述内容

項目 (英語名)	内容
適用範囲 (scope)	
対象 (level)	品質を記述する対象
対象の範囲 (extent)	品質を記述する対象の具体的な時空間範囲を記述
対象の説明 (levelDescription)	品質を記述する対象 (地名辞典を作成する際に引用した元データ等)に関する情報を記述
系譜 (Lineage)	
系譜の説明 (statement)	品質の適用範囲で記述したデータソースを、どのような手法によって整備したかについて記述

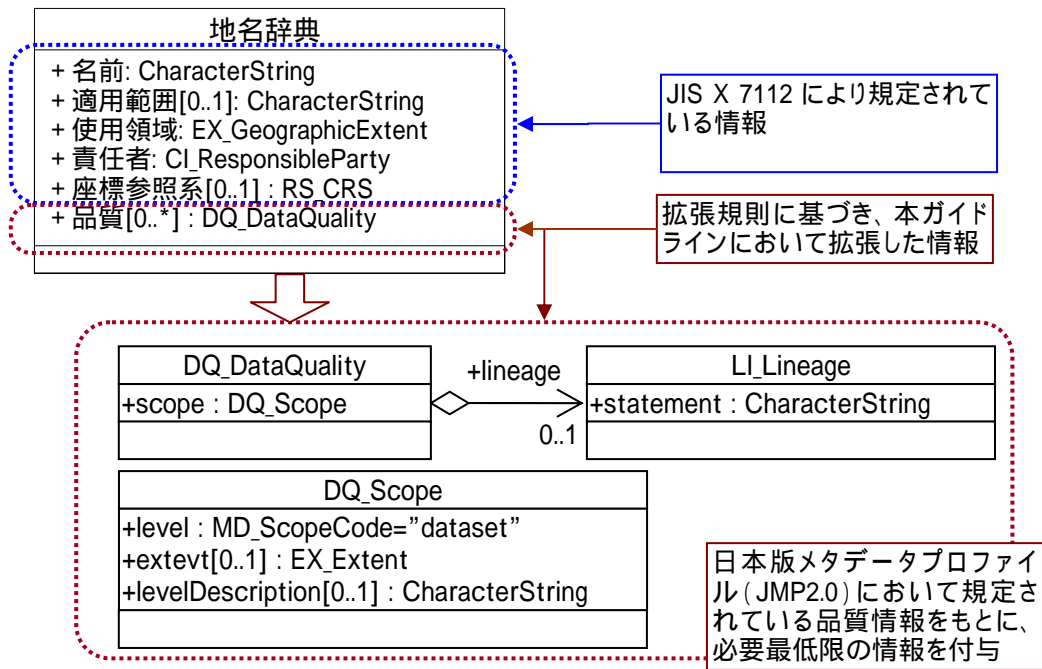


図 3-4 地名辞典に付与する品質情報

3-1-3. 地理識別子による空間参照系 (PI_LRS)

地理識別子による空間参照系は、場所型の集合により構成される、地理識別子を実世界に関連付ける参照系のことである。空間参照系には 2 種類あり、一つはこの地理識別子による空間参照系であり、もう一つは経緯度や平面直角座標系に代表される、座標による空間参照系である。例えば平面直角座標系の内容を示すために、この平面直角座標系の原点

がどこで、適用できる地理範囲がどこであるか等の情報が必要であるのと同じように、地理識別子による空間参照系においても、この空間参照系で利用できる場所型は何であり、適用できる地理範囲がどこであるか等の情報を示すことが必要である。

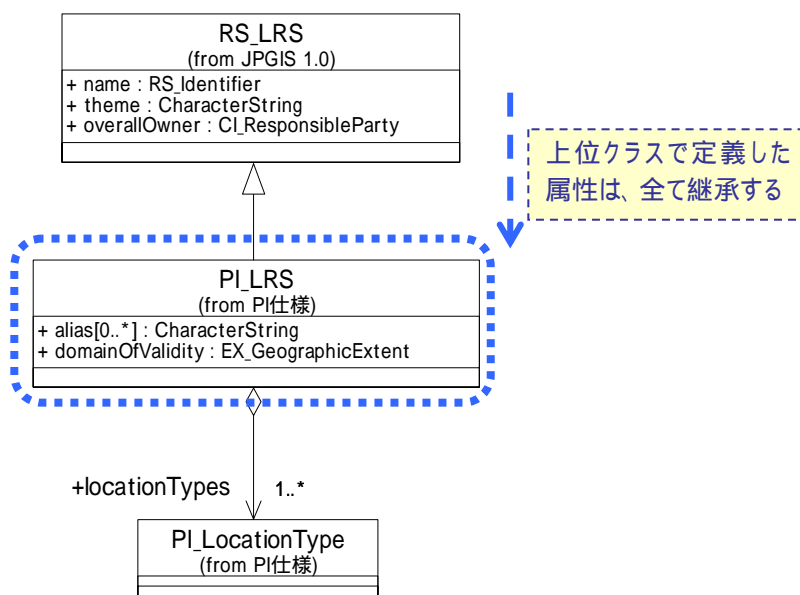


図 3-5 地理識別子による空間参照系 (PI_LRS)

表 3-5 地理識別子による空間参照系の属性及び関連役割情報

属性及び関連役割 (英語名)	記述条件	記述形式	概要及び定義
名前 (name)	必須	URI	地理識別子による空間参照系の名前をドメインによって記述
有効領域 (domainOfValidity)	任意記述	EX_GeographicExtent	空間参照系が適用する範囲を座標値、経緯度、地理識別子のいずれかで記述
主題 (theme)	必須	CharacterString	空間参照系について、その特徴(テーマや目的)を記述
全面的責任者 (overallOwner)	必須	CI_ResponsibleParty	空間参照系を管理する個人名、役職または組織名を記述
別名 (alias)	任意記述	CharacterString	空間参照系の別名を記述
場所型 (locationTypes)	必須	PI_LocationTypes	この空間参照系で利用できる場所型を記述

3-1-4. 場所インスタンス(PI_LocationInstance)

場所インスタンスとは、地理識別子と対応する位置情報との対をもったインスタンスのことであり、地理識別子の値や位置情報は、実際は場所インスタンスの中で記述される。場所インスタンスには、地理識別子と位置情報以外にも、代替地理識別子、時間範囲、管理者の属性情報を記述することができる。

また、場所インスタンスは、地理識別子の記述方法や定義を定めた場所型のインスタンスの位置付けであるため、必ず場所インスタンスのもととなる場所型を参照しなければならない。

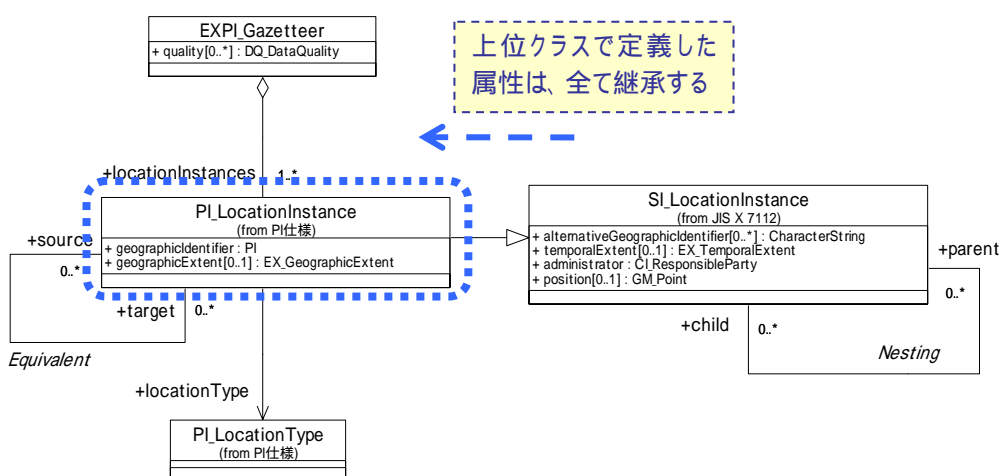


図 3-6 場所インスタンス (PI_LocationInstance)

表 3-6 場所インスタンスの属性及び関連役割情報

属性及び関連役割 (英語名)	記述条件	記述形式	概要及び定義
地理識別子 (geographicIdentifier)	必須	PI	一意に特定できる地理識別子の名称をPIの形式で記述
代替地理識別子 (alternativeGeographicIdentifier)	任意記述	CharacterString	同一の地物でも別の名称がある場合、上記の地理識別子と代替可能な別の識別子を記述
地理範囲 (geographicExtent)	任意記述	EX_GeographicExtent	地理識別子が取り得る範囲を記述する。範囲については、経緯度・座標値または、ポリゴンで記述
時間範囲 (temporalExtent)	任意記述	TM_Period	地理識別子の存続期間を記述 (実際の地物の設置期間を記述する。設置期間が不明な場合は、データ作成日を記述する。現在設置中の場合は、終了日は“Now”を記述)
管理者 (administrator)	必須	CI_ResponsibleParty	場所インスタンスの修正や削除に関して責任をもつ個人名、役職、または組織名を記述
位置 (position)	任意記述	GM_Point	地理識別子の代表点を記述
場所型 (locationType)	必須	PI_LocationType	場所インスタンスの型となる場所型を記述
地名辞典 (gazetteer)	任意記述	EXPI_Gazetteer	場所インスタンスが格納されている地名辞典を記述

3-1-5. 場所型 (PI_LocationType)

場所型とは、地理識別子の特性を定めた型のことである。3-1-1.で示したとおり、地理識別子は、PI タイプで定めた符号化規則に則って、PI として値が記述される。つまり、具体的に地理識別子を記述するのが場所インスタンスであるのに対し、その地理識別子のカテゴリや定義を記述したものが場所型となる。

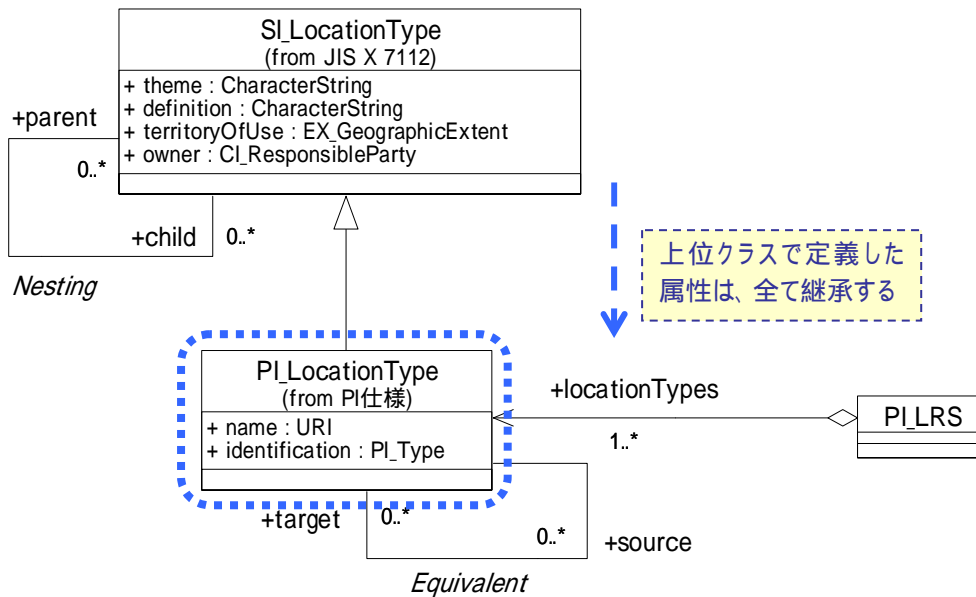


図 3-7 場所型 (PI_LocationType)

表 3-7 場所型の属性及び関連役割情報

属性及び関連役割 (英語名)	記述条件	記述形式	概要及び定義
名前 (name)	必須	URI	地理識別子の分類を表す場所型の名称を記述。名称は一意に識別できなければならない
主題 (theme)	必須	CharacterString	場所型の特徴(テーマや目的)を記述
識別 (identification)	必須	PI_Type	地理識別子の記述方法をPIタイプとして記述
定義 (definition)	必須	CharacterString	場所型の定義を記述
使用地域 (territoryOfUse)	必須	EX_GeographicExtent	場所型を使用する地域を記述
責任者 (owner)	必須	CI_ResponsibleParty	場所型の更新や削除に関して責任をもつ個人名、役職、または組織名を記述

3-2. データ作成時・運用時に守るべき規約

作成されるデータが多数のシステム間で相互に連携できることを担保するために、地名辞典等のデータを作成する際は、“3-1.で示したデータ構造に従うこと”をはじめ、守るべきルール（規約）がある。

規約は大きく PI や場所インスタンスの作成・運用に関する規約と空間参照系や場所型の作成・運用に関する規約の2つに分かれる。PI、場所インスタンス、地名辞典、地理識別子による空間参照系及び場所型を作成・運用する際は、以下に示す規約に従わなければならない。

3-2-1. PI 及び場所インスタンスの作成・運用時の規約

PI、場所インスタンス及び地名辞典を作成・更新・削除する場合は、以下に示す規約に従わなければならない。

(A)PI 及び場所インスタンス(PI_LocationInstance)の作成

- ✓ PI は、PI_Type に定義された応用スキーマ及び符号化規則に従って作成しなければならない。
- ✓ PI は、これをもつ場所インスタンス (PI_LocationInstance) が含まれる地名辞典 (EXPI_Gazetteer) の中で一意でなければならない。
- ✓ 場所インスタンス (PI_LocationInstance) は、3-1.で示すデータ構造に従って作成しなければならない。
- ✓ 場所インスタンス (PI_LocationInstance) は、自身が参照する PI_LocationType を必ず指定しなければならない。

(B)地名辞典(EXPI_Gazetteer)の作成

- ✓ 地名辞典 (EXPI_Gazetteer) は、3-1.で示すデータ構造に従って作成しなければならない。
- ✓ 地名辞典 (EXPI_Gazetteer) の名称は、地名辞典管理サービス (第5章で説明) の中で一意でなければならない。

(C)PI 及び場所インスタンス(PI_LocationInstance)の更新

- ✓ PI 及び場所インスタンス (PI_LocationInstance) の更新は、その場所インスタンスの管理者 (属性 administrator に記述されたもの) のみが行うことができる。

(D)地名辞典(EXPI_Gazetteer)の更新

- ✓ 地名辞典 (EXPI_Gazetteer) の更新は、その地名辞典の管理者 (属性 custodian に記述されたもの) のみが行うことができる。

(E)PI 及び場所インスタンス(PI_LocationInstance)の削除

- ✓ PI 及び場所インスタンス (PI_LocationInstance) の削除は、その場所インスタンスの管理者 (属性 administrator に記述されたもの) のみが行うことができる。

(F)地名辞典(EXPI_Gazetteer)の削除

- ✓ 地名辞典 (EXPI_Gazetteer) の削除は、その地名辞典の管理者 (属性 custodian に記述されたもの) のみが行うことができる。

3-2-2. 地理識別子による空間参照系及び場所型の作成・運用時の規約

地理識別子による空間参照系及び場所型を作成・更新・削除する場合は、以下に示す規約に従わなければならない。

(A)地理識別子による空間参照系(PI_LRS)の作成

- ✓ 地理識別子による空間参照系 (PI_LRS) は、3-1.で示すデータ構造に従って作成しなければならない。
- ✓ 地理識別子による空間参照系 (PI_LRS) は、ドメインをもたなければならない。
- ✓ 地理識別子による空間参照系 (PI_LRS) の名称にはドメイン名を記述し、他の空間参照系と識別できなければならない。

(B)場所型(PI_LocationType)の作成

- ✓ 場所型 (PI_LocationType) の名称は一意でなければならない。
- ✓ 場所型 (PI_LocationType) の中に含まれる PI タイプ (PI_Type) は、3-1.で示すデータ構造に従って作成しなければならない。
- ✓ 場所型 (PI_LocationType) は、3-1.で示すデータ構造に従って作成しなければならない。

(C)地理識別子による空間参照系(PI_LRS)の更新・削除

- ✓ 地理識別子による空間参照系 (PI_LRS) の更新は、その空間参照系の管理者 (属性 overallOwner に記述されたもの) のみが行うことができる。

(D)PI タイプ(PI_Type)及び場所型(PI_LocationType)の更新・削除

- ✓ PI タイプ (PI_Type) 及び場所型 (PI_LocationType) の更新は，その場所型の管理者 (属性 owner に記述されたもの) のみが行うことができる。

4. 地理識別子への位置情報の与え方

地理識別子は、対となる位置情報とセットで地名辞典内に「場所インスタンス」として格納されることで、初めて地図検索や情報検索における“キー”として利用することが可能となる。

この章では、地理識別子に付与する位置情報について、与え方や守るべき事項を示す。

4-1. 地理識別子に付与する位置情報の形状

地理識別子には、それに対応する位置情報を付与しなければならない。地理識別子への位置情報の付与は、実際には第3章で示したとおり、場所インスタンス(SI_LocationInstance)の属性“地理範囲”または“位置”に位置情報を記述することで行う。

位置情報を“地理範囲”で記述する場合、地理識別子の位置情報を、“地理範囲”として付与する場合、第3章で示したデータ構造に従い EX_GeographicExtent というクラスで地理範囲を示すこととなる。また、位置情報を“位置”で記述する場合は GM_Point というクラスで位置を示すこととなる。

4-1-1. “地理範囲”として位置情報を付与する方法

地理識別子の位置情報を、“地理範囲”として付与する場合、第3章で示したデータ構造に従い EX_GeographicExtent というクラスで地理範囲を示すこととなる。

EX_GeographicExtent とは、メタデータ(データの内容を説明するためのデータ)などで主に使用される、位置情報を記述クラスであり、以下の2つの中から1つを選択して、地理範囲を記述する。

(A) 地理範囲の境界をボックスで記述する場合

“地理範囲の境界をボックスで表す”とは、東西南北に直行した四角形で範囲を示すことをいう。この場合は、さらに範囲を緯度経度を用いて示す方法と、境界を座標値を用いて示す方法とがある。

緯度経度を用いて示す場合は、対象となる地理識別子を囲うボックスの対角線をなす2つの頂点の経度緯度を示すことになる。

座標値を用いて示す場合は、対象となる地理識別子を囲うボックスの対角線をなす2つの頂点の座標値を示すことになる。

表 4-1 地理範囲の境界をボックスで表す場合

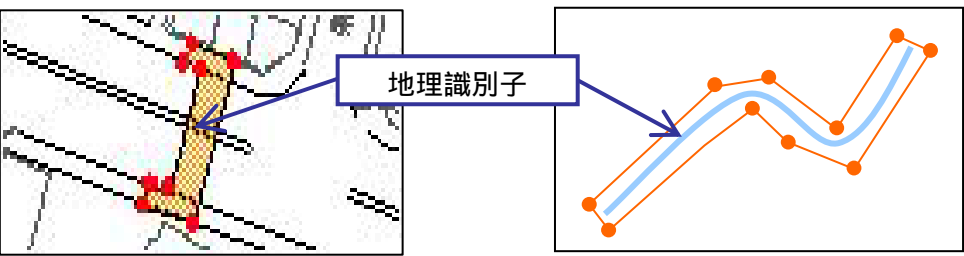
説明	実際の範囲に対して、東西南北に直交した矩形で与える。	
長所及び短所	長所	<ul style="list-style-type: none"> ・ 作成が比較的容易である。 ・ 表示範囲や検索範囲の設定が容易である。
	短所	<ul style="list-style-type: none"> ・ 狭長物の場合は不当に広い範囲を示すことがある。
具体的な付与方法	<p>地理識別子を囲うボックスの対角線をなす2つの頂点の座標を用いて4つの値を記述。</p> <p>【緯度経度で示す場合】</p> <p>2つの頂点の経度緯度から以下の4つの値を記述</p> <ul style="list-style-type: none"> ・ westBoundLongitude: 最西端の経度を記述 ・ eastBoundLongitude: 最東端の経度を記述 ・ southBoundLatitude: 最南端の緯度を記述 ・ northBoundLatitude: 最北端の緯度を記述 <p>【座標値で示す場合】</p> <p>2つの頂点の座標値から以下の4つの値を記述</p> <ul style="list-style-type: none"> ・ westBoundCoordinate: 最西端の X 座標値 ・ eastBoundCoordinate: 最東端の X 座標値 ・ southBoundCoordinate: 最南端の Y 座標値 ・ northBoundCoordinate: 最北端の Y 座標値 	

(B) 地理範囲の境界を多角形で表す場合

“地理範囲の境界を多角形で表す”とは、地理識別子の範囲を表4-2に示すような多角形を作成することで範囲を示すことをいう。

地理識別子が面としてデジタルデータ化されていれば、多角形を構成する各頂点の座標値をそのまま利用することができるが、図面等から施設や区域の地理識別子を多角形として範囲を示す場合は、デジタイズ等によって各頂点の座標値を付与することとなり、データ作成の作業が煩雑になる。

表 4-2 地理範囲の境界を多角形で表す場合

		
説明	実際の範囲に対して、境界多角形で与える。	
長所及び短所	長所	<ul style="list-style-type: none"> ・ 表示や検索等の利用性が向上する。 ・ 詳細な範囲を示すことができる。
	短所	<ul style="list-style-type: none"> ・ 点が多いと作成が複雑で困難となる。
具体的な付与方法	<ul style="list-style-type: none"> ・ 地理識別子が面としてデジタルデータ化されていれば、多角形を構成する各頂点の座標値をそのまま利用することができる。 ・ 鉄道、道路、下線などが線として表現されている場合にそれを多角形として位置情報を付与する場合も、同様にその地理識別子を囲うように別途多角形を作成する必要がある。 ・ 図面等に記載された施設や区域の地理識別子の範囲を作成する場合は、デジタイズ等によって、各頂点の座標値を取得する。 	

4-1-2. “位置”として位置情報を付与する方法

地理識別子の位置情報を、“位置”として付与する場合、第3章で示したデータ構造に従い GM_Point というクラスで地理範囲を示すこととなる。GM_Point とは、JPGIS で規定する点を表現するためのクラスであり、“位置”として付与する場合は、地理識別子の位置情報を代表点1点で表す。

この場合、代表点1点で位置情報を付与できるため、上述の“地理範囲”として付与する場合に比べて作業は容易である。しかし、地理識別子を1点で表すために、逆にその地理識別子の位置が分かりづらくなる場合や、画面上に表示する際の縮尺設定や、その地理識別子をキーにして検索する場合の検索範囲等を別途設定する必要がある場合もあり、目的に応じた位置情報の付与が重要である。

表 4-3 代表点で位置情報を付与する場合

		
説明	実際の範囲に対して、代表点 1 点で与える。	
長所及び短所	長所	<ul style="list-style-type: none"> ・ 作成が容易である。
	短所	<ul style="list-style-type: none"> ・ 1 点で表現するため、表示範囲や検索範囲の設定が別途必要となる。 ・ 実際の範囲が広い場合は一部しか示せない。
具体的な付与方法	<ul style="list-style-type: none"> ・ 地理識別子が点としてデジタルデータ化されていれば、その座標値をそのまま利用することができる。 ・ 地理識別子が線や面として表現されている場合にそれを代表点として位置情報を付与する場合は、空間演算等によって代表点を決定する必要がある。 ・ 図面等に記載された施設や区域の地理識別子の範囲を作成する場合は、デジタイズ等によって、代表点 1 点を取得する。 	

4-2. 位置情報を付与する際の座標系及び座標値の記述

建設情報を作成する際に利用する座標系は以下のものが多く利用されている。

(1) 平面直角座標

(2) 緯度経度

現行の電子納品要領・基準類においても、地理座標の種類として「平面直角座標」や「緯度経度(度、分、秒)」が規定されている。

本ガイドラインに従って地理識別子に位置情報を付与する際は、JPGIS1.0 で示された座標系を用いて付与することとする。JPGIS1.0 では、地理座標を表記する場合は、地理座標が参照する測地系と座標系を座標参照系として明記したうえで、その座標参照系に対応した座標値を記述することを明確に規定している。表 4-4 に、JPGIS1.0 の規定に準じた主な座標参照系識別子の表記方法を示す。

なお、測地の基準となる測地系には日本測地系と日本測地系 2000(世界測地系)があるが、本ガイドラインでは日本測地系 2000(世界測地系)を原則とする。

表 4-4 主な座標参照系識別子の表記方法

項目	表記方法	表記例		参照文献
座標参照系識別子	測地系の識別子と、座標系の識別子を「測地系 + “/” + 座標系」のように組み合わせて表記する。 【測地系】 日本測地系： TD 日本測地系 2000： JGD2000 【座標系】 平面直角座標系：系番号 + (X、Y) 緯度経度： (B、L)	日本測地系 2000 における緯度経度座標系	「JGD2000 / (B、L)」	地理情報標準プロファイル (JPGIS) Ver1.0
		日本測地系 2000 における平面直角座標系第 系	「JGD2000 / 6 (X、Y)」	
座標値 (平面直角座標)	平面直角座標(19系)を X 座標 (m) で記入 平面直角座標(19系)を Y 座標 (m) で記入	X 座標： -8298.6812m	「-8298.6812」	
		Y 座標： -34857.2912m	「-34857.2912」	
座標値 (緯度経度座標)	緯度を度(実数)で記入 経度を度(実数)で記入	北緯 35 度 02 分 13 秒	「35.0369」	
		東経 138 度 09 分 29 秒	「138.1581」	

【参考】緯度経度の表記方法（桁数）に対する実距離の目安

建設情報に位置座標を与える場合、記述する桁数や表記方法によってはそれが示す位置の正確度に影響する。緯度経度の表記方法および桁数の違いによる実距離の目安を以下に示す。

表 4-5 緯度経度の表記方法（桁数）に対する実距離の目安

緯度経度 (小数点以下桁数)	実距離	緯度経度 (小数点以下桁数)	実距離
0.001 度(3 桁)	92m	1 秒	30m
0.0001 度(4 桁)	9.2m	0.1 秒(1 桁)	3m
0.00001 度(5 桁)	92cm	0.01 秒(2 桁)	30cm
0.000001 度(6 桁)	9.2cm	0.001 秒(3 桁)	3cm
0.0000001 度(7 桁)	9.2mm	0.0001 秒(4 桁)	3mm
0.00000001 度(8 桁)	0.92mm	0.00001 秒(5 桁)	0.3mm

(岡山県の県庁所在地緯度を基準とした場合(「理科年表」より東京天文台編纂))

5. 地名辞典管理サービスの実装

作成した地名辞典を利用して、地理識別子をキーにした位置の検索等を行うためには、地名辞典管理サービスが必要となる。地名辞典管理サービスは地名辞典を管理するとともに、利用者の要求（=検索条件）に合致する地理識別子や位置情報を利用者に返答する機能をもつ。

また、地名辞典管理サービスは、地理識別子を介して他のサービスとも情報の連携ができるように、他のサービスからの要求に対しても処理を行い、合致する結果を要求してきたサービスに返答できるような機能をもつことが望ましい。そのためには、別途定める「[建設情報連携ポータル標準インタフェースガイドライン](#)」で示す関数仕様に従って機能を実装しなくてはならない。

なお、地名辞典管理サービスの詳細な機能や仕様等については、JIPDEC/DPC が公開している「[PI 詳細設計書](#)」を参照するとよい。

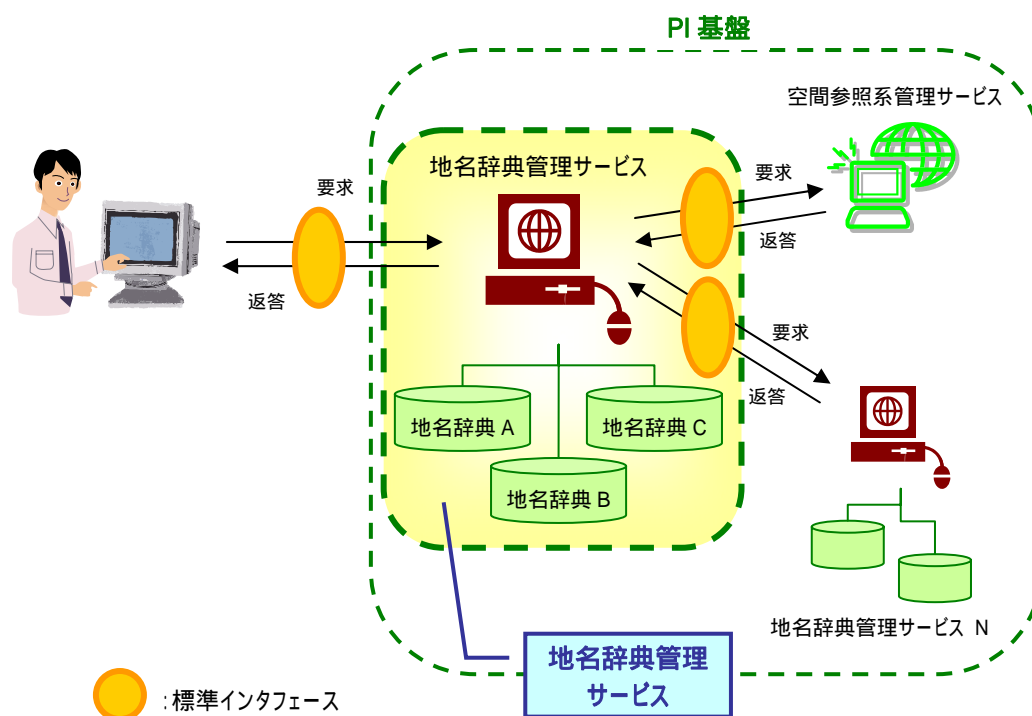


図 5-1 地名辞典管理サービスのイメージ

5-1. 地名辞典管理サービスの機能の概要

地名辞典管理サービスは、地名辞典の管理や、地理識別子の検索や変換に必要な情報を管理し、ユーザーからの要求に対して、適切な地理識別子を検索し、応答するサービ

スである。地名辞典管理サービスは、自身もつ地理識別子を検索することができなければならないが、他の地名辞典管理サービスもつ地理識別子を検索してもよい。

また、地名辞典管理サービスが提供すべき操作や管理すべき情報及び他の地名辞典管理サービスとの地理識別子のやり取りを行う場合に相互のサービス間で決定すべき事項を規則として定めている(5-1-2.)。地名辞典管理サービスはサービスを提供する際、この規則に従わなければならない。

地名辞典管理サービスは、3-1.に示すデータ構造に基づくデータを扱い、以下の機能を実現する。

表 5-1 地名辞典管理サービスの機能とその内容

機能	機能の内容
PI(地理識別子)の変換	インプットとなるPIに対し、指定された空間参照系に基づくPI、あるいは対応付く全てのPIを検索・取得する。 PIの変換は、自身が管理するPI_Gazetteerに含まれるPIのみならず、他のPI変換サービスが管理するPI_Gazetteerに含まれるPIを用いることができる。 地名辞典管理サービスは、分散環境下に存在しうる。
地名辞典の更新	地名辞典を更新する。
地名辞典の登録	地名辞典を新規に登録する。
地名辞典の提供	地名辞典変換サービスが扱う全ての地名辞典を提供する。
登録者の管理	場所インスタンスや地名辞典に登録するユーザの管理を行うことができる。
利用者の管理	PIの変換や地名辞典を利用するユーザの管理を行うことができる。

5-2. 地名辞典管理サービスの運用規約

地名辞典管理サービスを提供する運営主体は、他のサービスとデータや機能を相互に連携できるようにするために、サービスの運用にあたって守るべき規約が定められている。

本ガイドラインでは、定められた規約の概要を示す。規約の詳細な内容については、「[PI仕様書](#)」を参照するとよい。

なお、“地名辞典管理サービス”は、「PI仕様書」では“PI変換サービス”と呼ぶため、参照する際注意が必要である。

「PI仕様書」は以下のURLからダウンロードが可能である。

<http://www.dpc.jipdec.jp/gxml/contents/pi/index.html>

表 5-2 「PI 仕様書」で定める、地名辞典管理サービスの運用規約の概要

規約の項目	概要
データの管理	管理するデータは、「PI 仕様書」に示すデータ構造に従っていないとしない。
インターフェースの実装	地名辞典管理サービスは、「PI 仕様書」に示すインターフェースを実装しなくてはならない。
他の地名辞典管理サービスとの提携	他の地名辞典管理サービスがもつ地名辞典のデータを用いて変換を行うばあいは、その地名辞典管理サービスと提携を行い、データ利用の障害とならないようにすること。
サービスメタデータの実装	地名辞典管理サービスは、自身もつ地名辞典や機能をサービスメタデータとして実装し、ユーザからの要求に応じて応答できなければならない。
ユーザの管理	サービスやデータの信頼性を担保するために、地名辞典管理サービスは、地理識別子や地名辞典を登録するユーザを識別し、管理することが望ましい。

既存の、独自に整備した地名辞典を公開・共用させたい場合

現在、データ構造が JIS X 7112 や JPGIS の規定には準拠しないが、既存のアプリケーションで“住所検索”“目標物検索”“郵便番号検索”等として利用される独自に整備された地名辞典が、実際には数多く存在する。

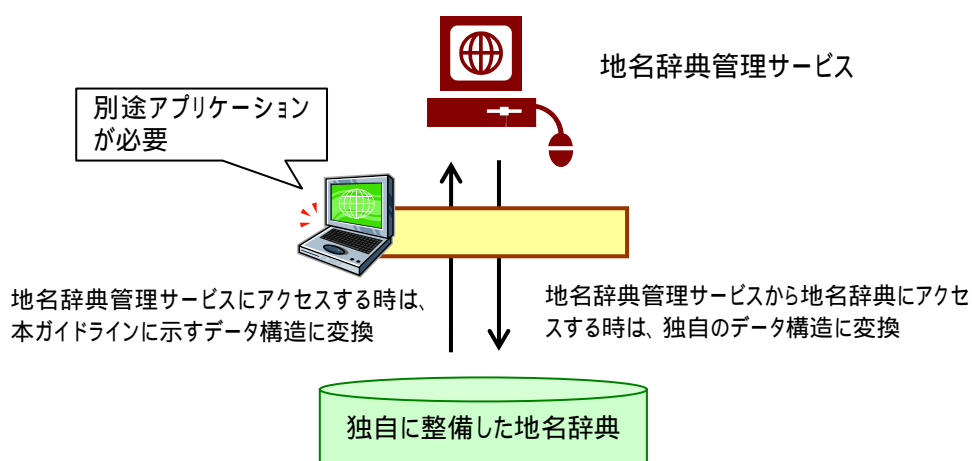
それらの地名辞典を地名辞典管理サービスを介して WWW 空間上で公開・共用させたい場合は、次のいずれかの方法を取る必要がある。

< 本ガイドラインで示すデータ構造に則るように、地名辞典を変換する >

独自に整備された地名辞典から必要な情報を抽出し、本ガイドラインで示すデータ構造に則るように地名辞典を変換することで、WWW 空間上で公開・共用させることができる。変換に際しては、別途変換ツールを作成するとよい。

< 簡易なアプリケーションを実装し、独自の地名辞典と地名辞典管理サービスを連携させる >

既存の地名辞典を本ガイドラインで示すデータ構造に変換することが困難な場合は、簡易なアプリケーションを実装し、地名辞典管理サービスにアクセスする時のみ本ガイドラインで示すデータ構造に変換すれば、運用上は WWW 空間上で公開・共用させることが可能である。もしこの方法を取る場合には、データ構造に加え規約についても本ガイドラインや「PI 仕様書」に示す規約に必ず従うように注意が必要である。



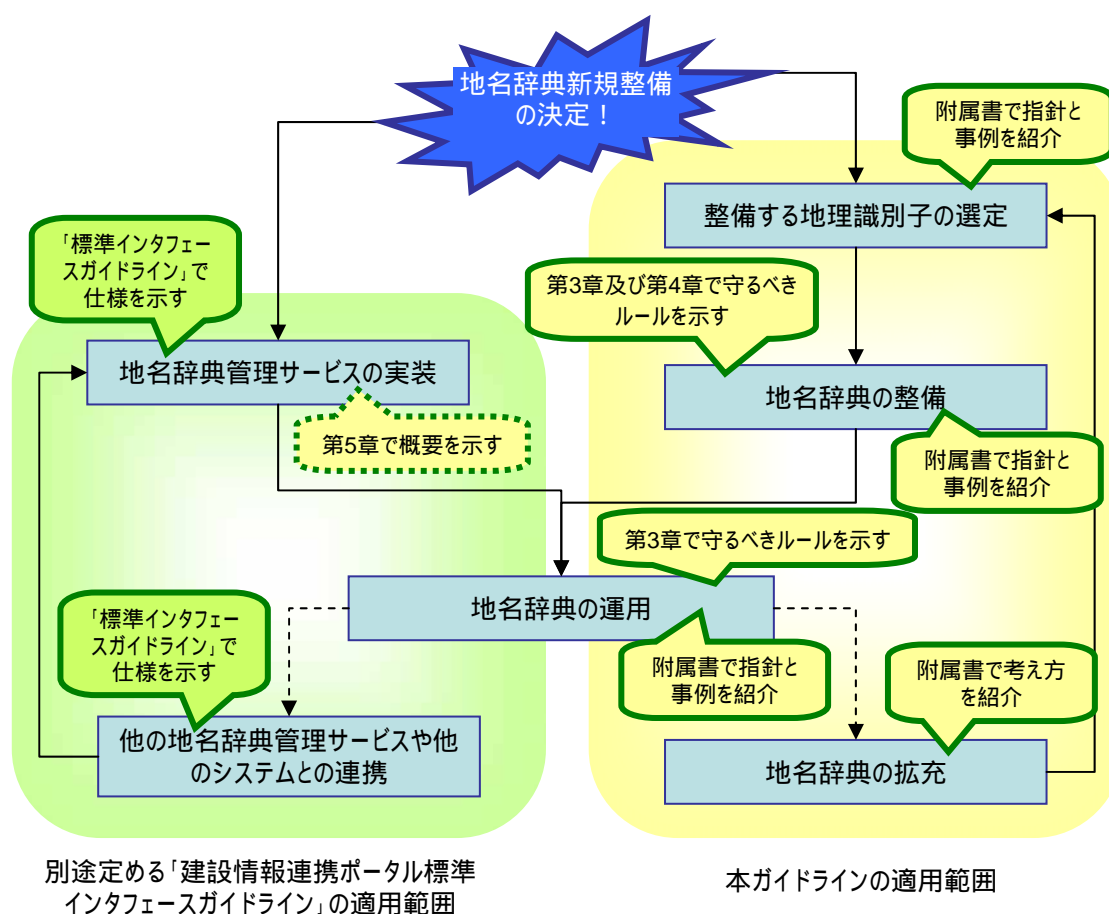
附属書「建設分野における地名辞典の整備・運用方法」

本附属書では、本編で示す地名辞典整備・運用のための決まりに則って、建設分野における地名辞典を整備するための指針及び方法を、事例を交えて解説する。

本附属書では道路分野及び河川分野を対象に、指針に沿って整備する地理識別子を整理し、整備をする方法について、事例を紹介する。また、下水道及び港湾分野についても、指針に沿った地理識別子選定の流れを簡潔に示す。

1. 建設分野における地名辞典の整備・運用の流れ

建設分野において地名辞典を整備・運用する場合のサイクルは、本編 1-3 で示した通り、以下のようなサイクルになると考えられる。



図附 1-1 地名辞典の整備・運用の流れ（再掲）

「整備する地理識別子の選定」の概説

地名辞典を整備するにあたり、業務分析を行ってどのような地理識別子が必要になるかを整理し、抽出する。さらに、整備する際に原典資料となる既存リソースにはどんなものがあるかを抽出する。抽出した地理識別子については、利用目的・利用頻度・整備のしやすさの点から評価を行い、整備優先度の高い地理識別子はどれかを整理する。

「地名辞典の整備」の概説

整理した優先度の高い地理識別子から、地理識別子を作成する。地理識別子の作成にあたっては、作成する機関が、既存リソースを管理しているかないかで、その手順は異なってくる。

「地名辞典の運用」の概説

地名辞典のデータは、GISにおける背景地図データと同様に、データの新鮮さを保つことが重要である。新鮮さを保つためには、整備した地名辞典に対して、施設や路線が新たに建設された場合や名称等が変更になった場合は随時地理識別子の追加や修正を行う必要がある。また、既存リソースの更新時期に合わせて、地名辞典のデータも定期的に更新をすることが必要である。

「地名辞典の拡充」の概説

地名辞典をより利用しやすくするために地理識別子の項目を増やす場合は、どの地理識別子を追加するかを選定するため、再び「整備する地理識別子の選定」に示す指針に基づき選定するとよい。

この附属書では、事例として道路分野、河川分野を取り上げ、地名辞典を新規に作成し、運用していく方法を、上に示した場面ごとに解説する。

2. 地名辞典の初期整備方法

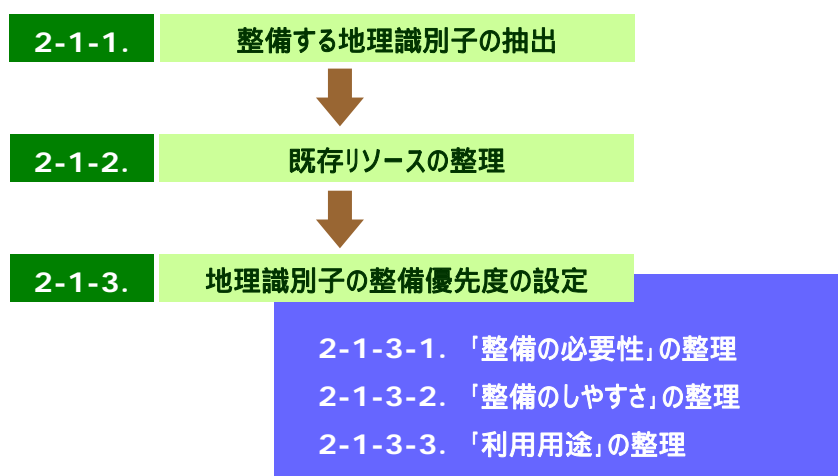
2-1. 整備する地理識別子の選定

ここでは、整備する地理識別子を選定する際の指針を示す。

指針には3つの段階があり、一つ目の段階は「候補となる地理識別子の抽出」、二つ目の段階が「既存リソースの抽出」、三つ目の段階が「地理識別子の整備優先度の設定」である。

「候補となる地理識別子の抽出」では、業務分析やヒアリング調査を行い、地名辞典を整備した場合に必要なであろう地理識別子は何なのかを抽出する。

「地理識別子の整備優先度の設定」では、上で整理した地理識別子に対して、整備の優先度を設定する。これは、上であげた地理識別子を費用面・工期面などで一度に整備することができない場合に、利用用途・利用頻度・整備のしやすさの面から地理識別子の評価を行い、優先的に整備すべき地理識別子を選定することで、目的に適った地理識別子を効率的に整備できるようにすることを目的としている。この附属書では、整備優先度を設定する方法を、指針として解説する。さらに、道路分野、河川分野で地理識別子として有効な12項目の地理識別子について、指針にしたがって整理した結果を示す。

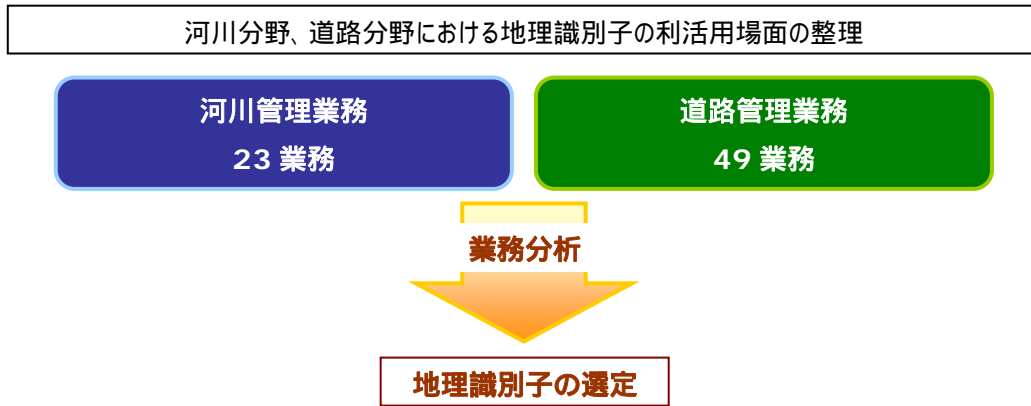


図附 2-1 整備する地理識別子の選定の流れ

2-1-1. 整備する地理識別子の抽出

地名辞典を整備するにあたり、まず、どのような場面で地名辞典を利用するか、利用場面を整理し、さらに、それぞれの利用場面でどのような地理識別子を必要とするかを抽出することが必要である。利用場面の整理や利用する地理識別子を抽出するには、業務分析やヒアリング調査の実施が有効である。

河川分野及び道路分野に関しては、ユースケース図を作成して業務分析を行い、地理識別子を利用する場面を整理し、共通して利用する地理識別子の種類を抽出した。



図附 2-2 河川分野、道路分野における地理識別子の整理

地理識別子を利用する場面を整理した結果として、河川分野及び道路分野では、以下の 12 種類の地理識別子が抽出された。

表附 2-1 河川分野及び道路分野で利用される地理識別子

	地理識別子名称	説明
1	住所	行政界及び海岸線によって構成される、地方自治法に定められた行政上の区域
2	管理河川	国土交通省の組織である河川事務所及び河川国道事務所が管轄する河川範囲
3	管理路線	一般国道のうち、政令(一般国道を指定する政令)で定められた国道の範囲
4	河川・距離標	堤防上に河口から延長方向に設置している杭や鉾
5	道路・距離標	道路起点からの延長を表示した標識で、道路端に設置してある杭や鉾
6	事務所管轄区域	国土交通省の組織である河川及び国道事務所が管轄する河川、路線範囲
7	出張所管轄区間	管理路線のうち、国土交通省の組織で事務所の出先機関である出張所が管轄する河川、路線で区切られた範囲
8	測点	路線測量において作成された計測点で、以下の点をいう。 ・道路中心線上に起点から等間隔で設置された点 ・道路中心線を構成する線形要素の変化点
9	交差点	十字路、T字路その他、2以上の車道が交わる部分。
10	水系	本川と本川に流れ込む支川を一連のまとまりとした範囲
11	流域	分水嶺を境界とする川の自然集水域
12	施設(道路・河川)	河川及び国道事務所が管理する河川管理施設と道路管理施設ならびに公共施設

2-1-2. 既存リソースの抽出

附属書 2-1-1.で抽出した整備対象の候補となる地理識別子について、整備の際に利用できる「既存リソース」を抽出する。

既存リソースには、

- ・ 電子化されているものと、されていないもの
- ・ 公共機関が作成したものと、民間企業が作成したもの
- ・ 自身の機関で所有しているものと、他機関が所有しているもの
- ・ 無償のものと、有償のもの

など、様々なパターンのリソースが存在すると考えられる。

道路分野・河川分野の地理識別子の整備に利用できる既存リソースは、以下のようなリソースが考えられる。

表附 2-2 河川分野、道路分野における地理識別子の既存リソース

地理識別子	既存リソース			
	名称()内は縮尺	整備範囲	データ名称	形状
河川・支川名	河川基盤地図 河川図(1/2, 500) 流域図(1/25, 000)	全国の一級河川直轄管理区間	河心線	線
	国土数値情報 数値地図(1/25, 000)	全国	流路	線
	数値地図(1/25, 000)	全国の都市計画区域外(都道府県、北海道支庁、郡市東京都の区、町村指定都市の区、大字町丁目、小字)	河川区間	線
	数値地図(1/2, 500)	全国の都市計画区域(都道府県、北海道支庁、郡市東京都の区、町村指定都市の区、大字町丁目、小字(街区))	河川区間	ポリゴン
	民間データ			
路線	DRM(縮尺 1/25, 000)	全国	基本道路リンクデータ	線
	道路管理システムデータ	全国		
	民間データ			
河川距離標	河川基盤地図 河川図(1/2, 500) 流域図(1/25, 000) 管内図	全国の一級河川直轄管理区間	杭	点
道路距離標	DRM(縮尺 1/25, 000) 管内図	全国	距離標位置データ	点
測点	設計図面			
水系	管内図	全国の一級河川直轄管理区間		
流域	管内図	全国の一級河川直轄管理区間		
交差点	DRM(縮尺 1/25, 000)	全国	交差点ノードデータ	代表点
河川施設	河川基盤地図 河川図(1/2, 500) 流域図(1/25, 000)	全国の一級河川直轄管理区間	河川構造物	ポリゴン
道路施設	DRM(縮尺 1/25, 000)	全国	施設等形状データ	ポリゴン
事務所管轄区域	管内図	全国の一級河川直轄管理区間		
出張所管轄区間	管内図	全国の一級河川直轄管理区間		
住所・地先				
都道府県・北海道の支庁レベル	街区レベル位置参照情報 (数値地図 2500 を基図として利用)	全国の都市計画区域(都道府県、市区町村、大字町丁目)		代表点
郡・市・東京都の区レベル				代表点
町・村・指定都市の区レベル	数値地図 2500/25000 (空間データ基盤) [地名]			代表点
大字・町丁レベル	民間データ			代表点

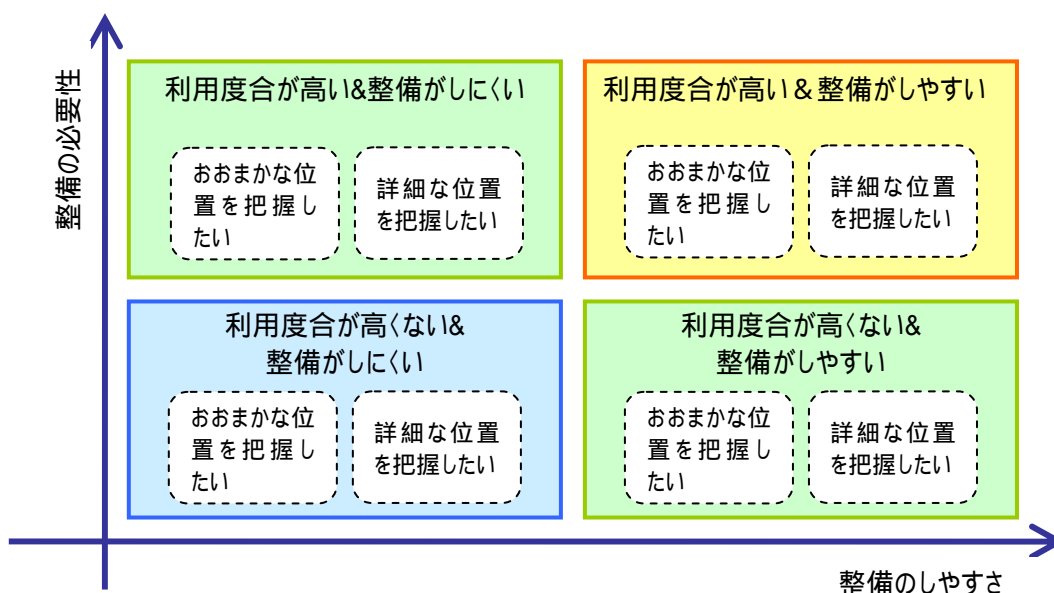
2-1-3. 地理識別子の整備優先度の設定

「地理識別子の整備優先度の設定」では、抽出した地理識別子に対して、整備の優先度を設定する。これは、抽出した地理識別子を費用面・工期面などで一度に整備することができない場合に、利用用途・利用頻度・整備のしやすさの面から地理識別子の評価を行い、優先的に整備すべき地理識別子を選定することで、目的に適った地理識別子を効率的に整備できるようにすることを目的としている。

整備優先度を設定するにあたっては、抽出した地理識別子に対して、

「整備の必要性」 「整備のしやすさ」 「利用用途」

の3つの視点から整理・検討を行い、整備優先度を設定する。



図附 2-3 整備優先度の設定による分類イメージ

2-1-3-1. 「整備の必要性」の整理

地理識別子は、全ての業務において一様に利用するというのではなく、多くの業務で利用されることが想定される地理識別子もあれば、逆に、ある特定の業務でのみ利用される地理識別子もある。そこで、ここではそれぞれの地理識別子がどれくらい多くの業務で利用されるかを整理する。整理の結果、「多くの業務で利用される地理識別子」と「特定の業務で利用される地理識別子」に分類することができ、その結果は、地名辞典の整備を検討する者にとって、地名辞典の整備優先度を設定するうえでの一つの有益な判断材料になる。

【整備方法の指針】

「整備の必要性」という観点から整理を行う際、以下の 2 つの視点から整理を行うことが必要である。

「 利用度合の高さ 」

地名辞典を用いたサービスを利用する際、多くの業務で利用される地理識別子はそれだけ汎用性が高く、利用度合の高い地理識別子とすることができる。ここでは地理識別子を利用度合の高さから整理することによって、地理識別子を「多くの業務で利用される地理識別子」と「特定の業務で利用される地理識別子」に分類し、地名辞典の整備の優先度を設定するうえでの判断材料を提供する。

「 非日常時の利用 」

の整理に加え、“非日常時(自然災害等による緊急時)にどの地理識別子を利用するか”という視点でも整理を行う。非日常時にも利用できるように整備するかどうかという問題は、地方整備局や事務所においては、重要なテーマであると考えられる。地理識別子を非日常時に利用することが想定されるものとそうでないものに整理することで、地名辞典を整備する者にとって一つの評価尺度、つまり優先度を提供することができる。

整理にあたっては、附属書 2-1-1.で整理した業務分析結果を基に、日常時の業務・非日常時の業務それぞれについて、どの業務でどの地理識別子を利用するかを整理することで、利用度合の高い地理識別子、非日常時に利用する地理識別子を抽出することができる。利用する業務数の大きい地理識別子は、地名辞典を利用するうえで利用度合の高い地理識別子と考えることができ、整備の必要性は高いと考えることができる。逆に、利用する業務数の少ない地理識別子は、ある特定の業務で利用される地理識別子と考えることができる。

この整理によって、地理識別子は、以下のように分類することができる。

表附 2-3 「整備の必要性」からみた地理識別子の分類

分類	利用度合	非日常時の利用
分類 A	高い	利用する
分類 B	高い	利用しない
分類 C	高くない	利用する
分類 D	高くない	利用しない

【河川分野・道路分野における整理結果】

整理方法の指針に従って河川分野・道路分野における地理識別子を整理した結果は、以下のようになる。

表附 2-4 河川分野・道路分野における地理識別子の整理結果

分類	該当する地理識別子	
	道路管理業務	河川管理業務
【分類 A】 (利用度合: 高い) (非日常時: 利用する)	住所 管理路線 道路・距離標 施設	住所 管理河川 施設
【分類 B】 (利用度合: 高い) (非日常時: 利用しない)		
【分類 C】 (利用度合: 高くない) (非日常時: 利用する)		河川・距離標
【分類 D】 (利用度合: 高くない) (非日常時: 利用しない)	交差点 出張所管轄区間 事務所管轄区域 測点	出張所管轄区間 事務所管轄区域 流域 水系

2-1-3-2. 「整備のしやすさ」の整理

それぞれの地理識別子の既存リソースは、「網羅性」「更新頻度」「データ利用に要する費用」がそれぞれ異なる。また、例えば民間データを活用した場合に発生することが予想される著作権等の問題など、データ作成にあたって生じうる課題もあり、既存リソースがもつ特性によって、整備のしやすさは大きく異なってくる。

ここでは、地理識別子を整備するにあたり、どのような流れで整理を行うと「整備の実現性(しやすさ)」という視点で整理を行うことができるかを示す。

【整備方法の指針】

整理項目には「既存リソースの有無」「デジタルデータかどうか」「公共機関作成のデータかどうか」「対象とするデータのデータ形状」の4項目を設定する。これらの整理項目によって整備のしやすさを整理することで、整備の上で懸念される点等を抽出することが可能となる。

【整理項目 : 既存リソースの有無】

地理識別子の中には、既存リソースとなる資料が存在しないものがあるかもしれない。

既存リソースが存在しない場合は、新たに既存リソースの選定作業や作成作業から行う必要があり、既存リソースがある場合に比べてより多くの工数を要することは明らかである。よって、既存リソースの無い地理識別子を整備することは、整備のしやすさが低いということができる。

【整理項目 : デジタルデータかどうか】

既存リソースがデジタルデータの場合、データ構造に差異はあるものの、基本的には地理識別子を抽出するためのプログラムを作成すれば、それを使用することで地理識別子を整備することが可能であると考えられる。しかし、既存リソースがデジタルデータでない場合は、資料の電子化（スキャン等による） マップデジタル化によるデータ入力 地理識別子データの作成、という工程が考えられ、既存リソースがデジタルデータである場合に比べてより多くの工数がかかる。よって、既存リソースがデジタルデータでない地理識別子を整備することは、整備のしやすさが低いということができる。

【整理項目 : 公共機関作成のデータかどうか】

既存リソースが公共機関が作成したデータの場合、基本的には無料でデータを利用することができるが、既存リソースが民間企業が作成したデータの場合は、著作権等の権利関係の問題やデータ使用料の問題等の発生が考えられ、これら諸問題をクリアすることが必要となる。

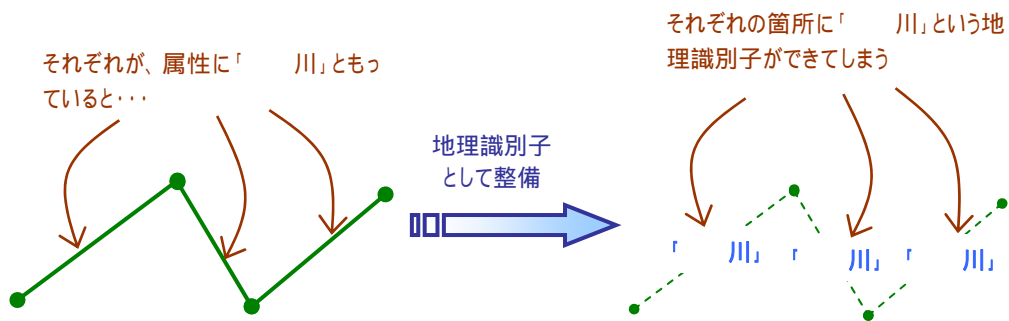
【整理項目 : 対象とするデータのデータ形状】

既存リソースでは地理識別子の整備対象となるデータが点・線・面のいずれかの形状で存在しているが、基となるデータの形状によっても地理識別子整備のしやすさに影響が出てくることが考えられる。

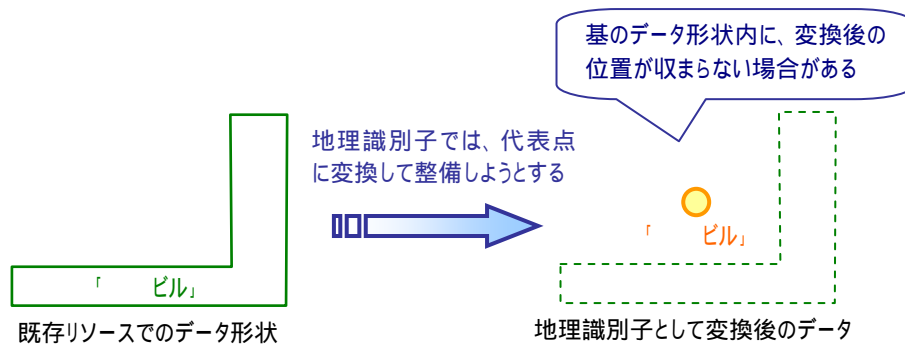
基となるデータの形状が「点」の場合は、特に整備のしやすさに影響を与える事項は存在しない。

基となるデータの形状が「線」の場合は、一本の道路や河川が多数の線データで作成されていることが多く、それぞれのデータが地理識別子となる名称を属性としてもっていることも多い。このような状況で対応策をとらないまま処理を行うと、図附 2-4 にあるように、同じ値をもつ地理識別子が多数存在してしまうことになるので、別途重複する地理識別子の統合処理を行う等の対応が必要である。

基となるデータの形状が「面」の場合は、地理識別子がもる地理範囲をそのまま「面」とする場合は特に問題となる事項は存在しないと考えられるが、例えば、地理識別子では地理範囲を「点」として代表点をもたせようとする、処理の結果、図附 2-5 にあるように、基のデータ形状の範囲外に代表点が位置してしまうこともよくある。この場合についても、基の形状内に代表点が納まるように対応をする必要がある。

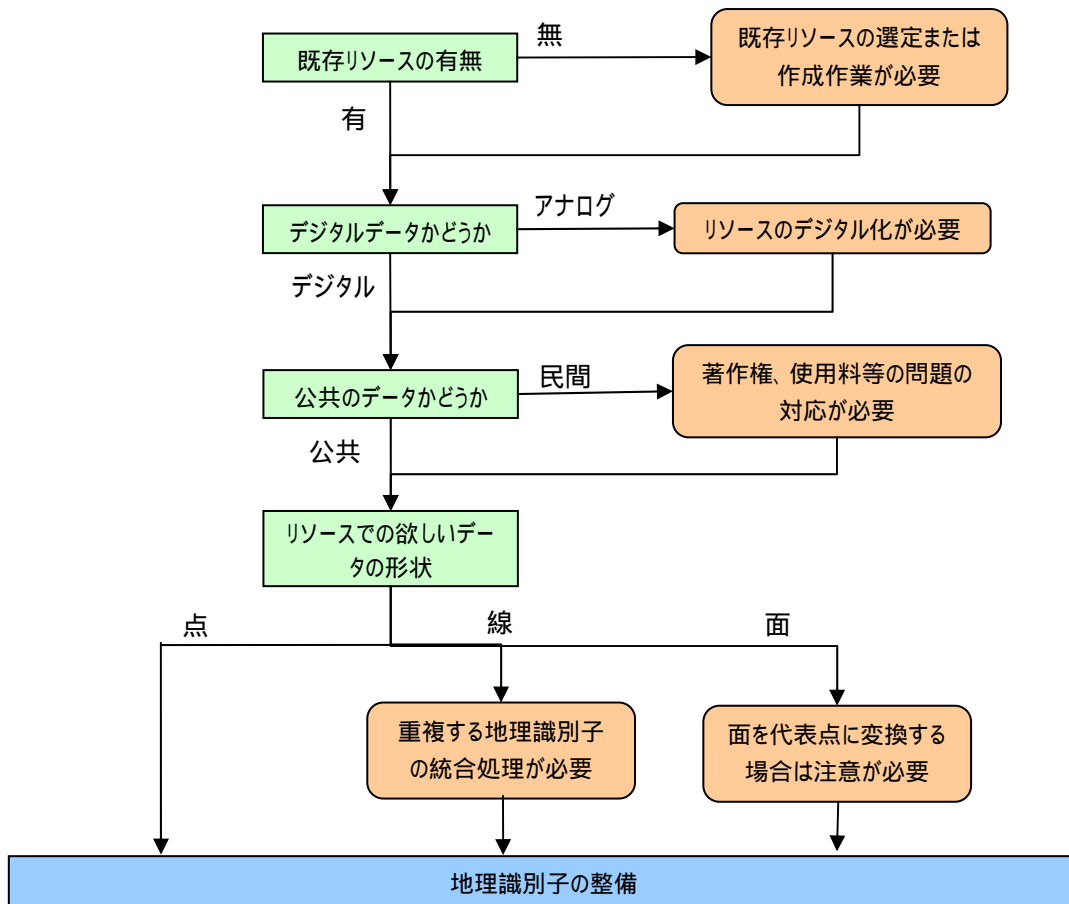


図附 2-4 既存リソースでのデータの形状が「線」の時に生じやすい問題



図附 2-5 既存リソースでのデータ形状が「面」の時に生じやすい問題

整理項目 ~ をフローの形式で示すと、図附 2-6 のようになる。



図附 2-6 整備のしやすさから見た、優先度設定の整理フロー

【河川分野・道路分野における整理結果】

整理方法の指針に従って河川分野・道路分野における地理識別子を整理した結果は、以下のようになる。

表附 2-5 整備のしやすさから見た、地理識別子の整理結果

地理識別子名	項目 既存リソース名	項目 デジタル/ アナログ	項目 公共/ 民間	項目 形状	注意点
整備のしやすい地理識別子とリソース (項目 : リソース有、項目 : デジタル、項目 : 公共、項目 : 点)					
河川距離標	河川基盤地図	デジタル	公共	点	
道路距離標	DRM	デジタル	公共	点	
交差点	DRM	デジタル	公共	点	
住所	街区レベル位置参照情報	デジタル	公共	点	
	数値地図	デジタル	公共	点	
整備の際注意が必要な地理識別子とリソース (項目 : リソース有、項目 : デジタル、項目 : 公共、項目 : 線または面) (項目 : リソース有、項目 : デジタル、項目 : 民間)					
施設(道路)	DRM	デジタル	公共	面	ポリゴンの代表点を地理識別子をもつ地理範囲にする場合
施設(河川)	河川基盤地図	デジタル	公共	面	
河川	数値地図	デジタル	公共	面	
河川	河川基盤地図	デジタル	公共	線	地理識別子の統合処理
	国土数値情報	デジタル	公共	線	
	数値地図	デジタル	公共	線	
路線	DRM	デジタル	公共	線	
	民間データ	デジタル	民間		著作権等の権利関係 データ使用料
整備がしづらい地理識別子とリソース (項目 : リソース有、項目 : アナログ)					
測点	設計図面	アナログ	公共		電子化が必要
水系	河川管内図	アナログ	公共		
流域	河川管内図	アナログ	公共		
河川距離標	河川管内図	アナログ	公共		
道路距離標	道路管内図	アナログ	公共		
事務所管轄区域	管内図(道路・河川)	アナログ	公共		
出張所管轄区間	管内図(道路・河川)	アナログ	公共		

2-1-3-3. 「利用用途」の整理

ここでは、附属書 2-1-1.で行った業務分析結果を基に、地名辞典の整備優先度を設定するために、それぞれの地理識別子がどのような用途で利用されるかを整理する。

【整理方法の指針】

それぞれの地理識別子の利用用途は、地理識別子をもつ地理範囲によって大きく「おおまかな位置を把握したい場合」「詳細な位置を把握したい場合」の2種類に分類することができると思われる。

「おおまかな位置を把握したい場合」は、“位置の特定”や“建設情報の登録”のために地名辞典を利用する場面では、対象箇所と地理識別子をもつ地理範囲との位置関係を把握するために利用される。管轄内の地理に事務所や出張所の職員は詳しいと想定できるため、まずおおまかな位置関係を把握することができれば、対象箇所の詳細な位置はそこから判断することが容易にできるかもしれない。また、“建設情報の検索”のために地名辞典を利用する場面では、管理路線や管理河川を含むおおまかな地理範囲を設定し、その中に含まれる建設情報を検索することで、一度に多くの建設情報を取得することが可能になる。

一方、「詳細な位置を把握したい場合」は、“位置の特定”“建設情報の登録”“建設情報の検索”のいずれの利用場面においても、狭い地理範囲をもつ地理識別子を利用することでダイレクトに位置を特定することができる。問い合わせ対応のために対象箇所の位置を特定したり、各種の履歴を登録する場面においては、ダイレクトに位置を特定できた方が、地図画面を拡大・縮小・ドラッグしながら詳細な位置を特定するよりも簡単に、早く位置を特定することができる。

【河川分野・道路分野における整理結果】

上記の方針に従って地理識別子を分類すると、以下のように分類することができる。

表附 2-6 利用用途から見た、地理識別子の分類

「おおまかな位置を把握したい場合」	「詳細な位置を把握したい場合」
住所(都道府県レベル、市区町村レベル)	住所(大字・町丁目レベル)
管理河川	河川距離標
管理路線	道路距離標
事務所管轄区域	測点
出張所管轄区間	交差点
水系	施設
流域	

2-1-3-4. 地理識別子の整備優先度の設定

附属書 2-1-3-1. ~ 2-1-3-3. で整理した結果を図附 2-3 に示した図に適用すると、整備優先度の高い地理識別子を設定することができる。

【河川分野・道路分野における整理結果】

整理方法の指針に従って河川分野・道路分野における地理識別子を整理した結果は、以下のようになる。

< 河川分野の場合 >

整備主体	用途	優先度の高い地理識別子	リソース
河川事務所	おおまかな位置を把握したい	住所	街区レベル位置参照情報 数値地図
		管理河川	河川基盤地図
	詳細な位置を把握したい	住所	街区レベル位置参照情報 数値地図
		施設	河川基盤地図(河川)

< 道路分野の場合 >

整備主体	用途	優先度の高い地理識別子	リソース
道路事務所	おおまかな位置を把握したい	住所	街区レベル位置参照情報 数値地図
		管理路線	DRM
	詳細な位置を把握したい	住所	街区レベル位置参照情報 数値地図
		施設	DRM(道路)
		道路・距離標	DRM

2-2. 地理識別子による空間参照系及び場所型の作成

地理識別子を作成するにあたっては、地理識別子の種類の内容や記述ルールを定めた場所型や地理識別子による空間参照系の作成が必要である。

2-2-1. 地理識別子による空間参照系の作成

例えば「平面直角座標系」等の座標による空間参照系では、その内容として原点の位置など、その空間参照系を説明する情報が必要であるように、地理識別子による空間参照系でも、その空間参照系を説明する情報が必要である。

地理識別子による空間参照系は、本編 3-1.で示したデータ構造に従ってデータを作成する。ただし、地理識別子による空間参照系は地方整備局や事務所の単位で必要という訳ではなく、分野ごとに一つ地理識別子による空間参照系を定義し、各地方整備局や事務所はそれを利用すればよい。

地理識別子による空間参照系の作成例

地理識別子による空間参照系の作成方法の検討

地理識別子による空間参照系の作成例を、表附 2-7 に示す。

表附 2-7 地理識別子による空間参照系の作成例

属性・関連役割	値		
名前	road.srs.jacic.or.jp		
地理範囲	日本全国		
主題	道路分野で建設情報を連携させる際に用いる地理識別子による空間参照系		
全面的責任者	電子地図/建設情報連携小委員会		
別名	道路分野に関する空間参照系		
場所型	住所型	管理路線型	道路距離標型
	交差点型	測点型	事務所管轄区域型
	出張所管轄区域型	施設型	

地理識別子による空間参照系のサンプル

作成した、地理識別子による空間参照系のサンプルを、表附 2-8 に示す。

実際のデータの形式は空間参照系管理サービスに依存するが、ここでは例として XML 形式での作成例を示す。

表附 2-8 地理識別子による空間参照系のサンプル (XML 形式)

以下、地理識別子による空間参照系のインスタンス部分の抜粋

```
<jci:PI_LRS id="lrs_road">
  <jps:RS_LRS.name>road.srs.jacic.or.jp</jps:RS_LRS.name>
  <jps:RS_LRS.theme>道路分野で建設情報を連携させる際に用いる地理識別子による空間参照系</jps:RS_LRS.theme>
  <jps:RS_LRS.overallOwner>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>電子地図/建設情報連携小委員会
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:RS_LRS.overallOwner>
  <jps:RS_LRS.locationTypes idref="lt_add">
  <jps:RS_LRS.locationTypes idref="lt_rd">
  <jps:RS_LRS.locationTypes idref="lt_rdkp">
  <jps:RS_LRS.locationTypes idref="lt_its">
  <jps:RS_LRS.locationTypes idref="lt_sp">
  <jps:RS_LRS.locationTypes idref="lt_ofd">
  <jps:RS_LRS.locationTypes idref="lt_bofs">
  <jps:RS_LRS.locationTypes idref="lt_fac">
  <jci:PI_LRS.alias>道路分野に関する空間参照系</jci:PI_LRS.alias>
  <jci:PI_LRS.domainOfValidity>日本全国</jci:PI_LRS.domainOfValidity>
</jci:PI_LRS>
```

2-2-2. 場所型の作成

地理識別子をどのように記述するかは、場所型を定義し、その中で記述しなければならない。場所型は、本編 3-1 で示したデータ構造に従って作成されなければならない。また、本編 3-2 で示した作成の際の規約にも従わなければならない。

ここでは、附属書 2-1 で抽出した 12 種類の地理識別子について、場所型の作成例を示す。

2-2-2-1. 住所型

場所型の作成方法の検討

「住所型」の場所型の作成例は、表附 2-9 のようになる。また、地理識別子の記述方法を定める住所型の PI タイプの作成例を、別途示す。

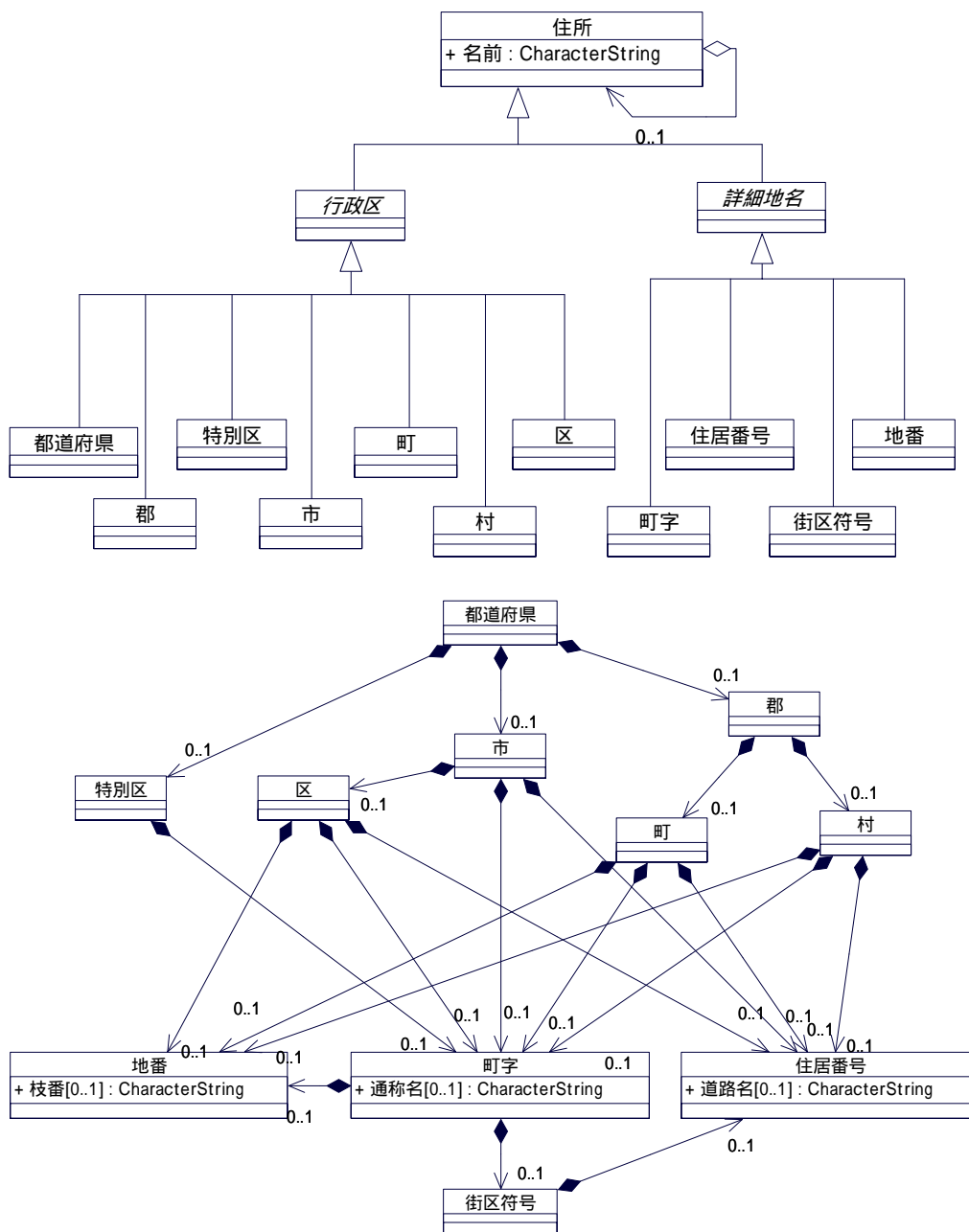
表附 2-9 「住所型」の場所型の作成例

属性・関連役割	値
名前	住所型
主題	住所の位置の指定
識別	(別途掲載)
定義	行政界及び海岸線によって構成される、地方自治法に定められた行政上の区域
使用地域	日本全国
責任者	電子地図/建設情報連携小委員会

住所型のPI タイプ例

応用スキーマ

住所型の応用スキーマを、図附 2-7 及び表附 2-10 に示す。



図附 2-7 住所型の応用スキーマ

表附 2-10 住所型の応用スキーマ定義文書

型	名前	住所	上位となる型	なし	
	定義	生活の本拠であって、法律関係进行处理する場合の基準となる場所。			
	抽象/具象区分	具象			
	属性	属性名	名前		
		定義	住所を識別する値		
		型	CharacterString		
		多重度	1	順序指標	なし
	役割	役割名	要素		
		定義	住所を構成する行政区や詳細地名		
型		住所			
多重度		0..1	順序指標	なし	
型	名前	行政区	上位となる型	住所	
	定義	行政上のまとまり。実体は下位の型のいずれかとなる。			
	抽象/具象区分	抽象			
型	名前	都道府県	上位となる型	行政区	
	定義	日本における行政区分の一つで、地方自治法に定める普通地方公共団体である都道府県。以下の役割のいずれかをもつ。			
	抽象/具象区分	具象			
	役割	役割名	特別区		
		定義	東京都内の特別区への参照。都道府県の名前が「東京都」の場合にのみ用いることができる。		
		型	特別区		
		多重度	0..1	順序指標	なし
	役割	役割名	郡		
		定義	都道府県内の郡		
		型	郡		
		多重度	0..1	順序指標	なし
	役割	役割名	市		
		定義	都道府県内の市		
		型	市		
		多重度	0..1	順序指標	なし
	型	名前	特別区	上位となる型	行政区
定義		地方自治法第 281 条第 1 項に規定する「都の区」。			
抽象/具象区分		具象			
役割		役割名	町字		
		定義	特別区を構成する町字		
		型	町字		
		多重度	0..1	順序指標	なし
型		名前	郡	上位となる型	行政区
	定義	地方行政区画の一種。住所表記や、広域連合体(広域行政圏)の範囲、都道府県議会選挙区の区割などに用いられる。以下のいずれかの役割をもつ。			
	抽象/具象区分	具象			
	役割	役割名	町		
		定義	郡内の町。		

	型	町			
	多重度	0.1	順序指標	なし	
	役割	役割名	村		
		定義	郡内の村。		
		型	村		
多重度	0.1	順序指標	なし		
型	型	名前	市	上位となる型	行政区
		定義	地方自治法第八条に示される市。日本における基礎自治体の単位。区、地番、町字、住居番号のいずれかをもつことができる。		
		抽象/具象区分	具象		
	役割	役割名	区		
		定義	区への参照。政令指定都市の場合に用いる。		
		型	区		
		多重度	0.1	順序指標	なし
	役割	役割名	地番		
		定義	市内の地番。		
		型	地番		
		多重度	0.1	順序指標	なし
	役割	役割名	町字		
		定義	市内の町字。		
		型	町字		
		多重度	0.1	順序指標	なし
	役割	役割名	住居番号		
		定義	市内の住居番号。道路名を記述する。		
型		住居番号			
多重度		0.1	順序指標	なし	
型	型	名前	町	上位となる型	行政区
		定義	地方自治法第八条に示される町。日本における基礎自治体の単位。地番、町字、住居番号のいずれかをもつことができる。		
		抽象/具象区分	具象		
	役割	役割名	地番		
		定義	市内の地番。		
		型	地番		
		多重度	0.1	順序指標	なし
	役割	役割名	町字		
		定義	市内の町字。		
		型	町字		
		多重度	0.1	順序指標	なし
	役割	役割名	住居番号		
		定義	市内の住居番号。道路名を記述する。		
		型	住居番号		
		多重度	0.1	順序指標	なし
	型	名前	村	上位となる型	行政区
定義		地方自治法第八条に示される村。日本における基礎自治体の単位。地番、町字、住居番号のいずれかをもつことができる。			

	抽象/具象区分	具象		
役割	役割名	地番		
	定義	市内の地番。		
	型	地番		
	多重度	0..1	順序指標	なし
役割	役割名	町字		
	定義	市内の町字。		
	型	町字		
	多重度	0..1	順序指標	なし
役割	役割名	住居番号		
	定義	市内の住居番号への参照。道路名を記述する。		
	型	住居番号		
	多重度	0..1	順序指標	なし
型	名前	区	上位となる型	行政区
	定義	地方行政区画の一種。住所表記や、広域連合体(広域行政圏)の範囲,都道府県議会選挙区の区割などに用いられる。地番,町字または,住居番号のいずれかをもつことができる。		
	抽象/具象区分	具象		
	役割	役割名	地番	
定義		市内の地番。		
型		地番		
多重度		0..1	順序指標	なし
役割	役割名	町字		
	定義	市内の町字。		
	型	町字		
	多重度	0..1	順序指標	なし
役割	役割名	住居番号		
	定義	市内の住居番号。道路名を記述する。		
	型	住居番号		
	多重度	0..1	順序指標	なし
型	名前	町字	上位となる型	詳細地名
	定義	地方行政区画の一種。住所表記や、広域連合体(広域行政圏)の範囲,都道府県議会選挙区の区割などに用いられる。地番または街区符号をもつことができる。		
	抽象/具象区分	具象		
	役割	役割名	地番	
定義		市内の地番。		
型		地番		
多重度		0..1	順序指標	なし
役割	役割名	街区符号		
	定義	町字内の街区符号。		
	型	街区符号		
	多重度	0..1	順序指標	なし
型	名前	地番	上位となる型	詳細地名
	定義	明治32年に制定された不動産登記法により定められた土地の番号		
	抽象/具象区分	具象		

属性	属性名	枝番		
	定義	分筆された場合の補助番号		
	型	CharacterString		
	多重度	0..1	順序指標	なし
型	名前	街区符号	上位となる型	詳細地名
	定義	道路や鉄道などの物理的な境界によって区切られた区画「街区」に割り当てられた番号。		
	抽象/具象区分	具象		
属性	属性名	通称名		
	定義	京都市において用いられる道路の名前に「東入る」「西入る」「上る」「下る」を付けたもの。		
	型	CharacterString		
	多重度	0..1	順序指標	なし
役割	役割名	住居番号		
	定義	街区内の住居番号。		
	型	住居番号		
	多重度	0..1	順序指標	なし
型	名前	住居番号	上位となる型	詳細地名
	定義	建物の出入口の位置に応じ、道路等に沿って10メートル間隔でつけた番号。		
	抽象/具象区分	具象		
属性	属性名	道路名		
	定義	市町村内の道路の名前及び当該道路に接し、又は当該道路に通ずる通路を有する建物その他の工作物につけられる住居番号を用いて表示する方法により住所が定められた場合の道路名前。		
	型	CharacterString		
	多重度	0..1	順序指標	なし

符号化規則

図附 2-7 に示した応用スキーマを符号化するためのタグ一覧を、表附 2-11 に示す。

表附 2-11 住所型の応用スキーマのためのタグ一覧

クラス	属性	タグ
住所		Jusho
	名前	name
	行政区	Gyoseiku
	都道府県	Todofuken
	特別区	Tokubetsuku
	郡	Gun
	市	Shi
	町	Machi
	村	Mura
	区	Ku
	町字	Choaza
	詳細地名	Shosaichimei
	街区符号	Gaikufugo
	住居番号	Jyukyobango
	道路名	doromei
	地番	Chiban
	枝番	edaban

住所型の符号化規則を、表附 2-12 に示す。

表附 2-12 住所型の符号化規則

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Jyusho" type="Jyusho"/>
  <xs:element name="Todofuken" type="Todofuken" substitutionGroup="Jyusho"/>
  <xs:element name="Tokubetsuku" type="Tokubetsuku" substitutionGroup="Jyusho"/>
  <xs:element name="Shi" type="Shi" substitutionGroup="Jyusho"/>
  <xs:element name="Gun" type="Gun" substitutionGroup="Jyusho"/>
  <xs:element name="Ku" type="Ku" substitutionGroup="Jyusho"/>
  <xs:element name="Machi" type="Machi" substitutionGroup="Jyusho"/>
  <xs:element name="Mura" type="Mura" substitutionGroup="Jyusho"/>
  <xs:element name="Chiban" type="Chiban" substitutionGroup="Jyusho"/>
  <xs:element name="Choaza" type="Choaza" substitutionGroup="Jyusho"/>
  <xs:element name="Jyukyobango" type="Jyukyobango" substitutionGroup="Jyusho"/>
  <xs:element name="Gaikufugo" type="Gaikufugo" substitutionGroup="Jyusho"/>
  <xs:complexType name="Jyusho">
    <xs:sequence>
      <xs:element ref="Jyusho" minOccurs="0"/>
    </xs:sequence>
    <xs:attributeGroup ref="IM_ObjectIdentification"/>
    <xs:attribute name="name" type="CharacterString"/>
  </xs:complexType>
  <xs:complexType name="Gyoseiku">
    <xs:complexContent>
      <xs:extension base="Jyusho"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="Shosaichimei">
    <xs:complexContent>
      <xs:extension base="Jyusho"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="Todofuken">
    <xs:complexContent>
      <xs:extension base="Gyoseiku">
        <xs:sequence>
          <xs:element name="Tokubetsuku" type="Tokubetsuku" minOccurs="0"/>
          <xs:element name="Shi" type="Shi" minOccurs="0"/>
          <xs:element name="Gun" type="Gun" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="Gun">
    <xs:complexContent>
      <xs:extension base="Gyoseiku">
        <xs:sequence>
          <xs:element name="Machi" type="Machi" minOccurs="0"/>
          <xs:element name="Mura" type="Mura" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="Machi">

```

```

<xs:complexContent>
  <xs:extension base="Gyoseiku">
    <xs:sequence>
      <xs:element name="Chiban" type="Chiban" minOccurs="0" />
      <xs:element name="Choaza" type="Choaza" minOccurs="0" />
      <xs:element name="Jyukyobango" type="Jyukyobango" minOccurs="0" />
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="Mura">
  <xs:complexContent>
    <xs:extension base="Gyoseiku">
      <xs:sequence>
        <xs:element name="Chiban" type="Chiban" minOccurs="0" />
        <xs:element name="Choaza" type="Choaza" minOccurs="0" />
        <xs:element name="Jyukyobango" type="Jyukyobango" minOccurs="0" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="Shi">
  <xs:complexContent>
    <xs:extension base="Gyoseiku">
      <xs:sequence>
        <xs:element name="Ku" type="Ku" minOccurs="0" />
        <xs:element name="Chiban" type="Chiban" minOccurs="0" />
        <xs:element name="Choaza" type="Choaza" minOccurs="0" />
        <xs:element name="Jyukyobango" type="Jyukyobango" minOccurs="0" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="Ku">
  <xs:complexContent>
    <xs:extension base="Gyoseiku">
      <xs:sequence>
        <xs:element name="Chiban" type="Chiban" minOccurs="0" />
        <xs:element name="Choaza" type="Choaza" minOccurs="0" />
        <xs:element name="Jyukyobango" type="Jyukyobango" minOccurs="0" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="Tokubetsuku">
  <xs:complexContent>
    <xs:extension base="Gyoseiku">
      <xs:sequence>
        <xs:element name="Chiban" type="Chiban" minOccurs="0" />
        <xs:element name="Choaza" type="Choaza" minOccurs="0" />
        <xs:element name="Jyukyobango" type="Jyukyobango" minOccurs="0" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>

```

```
</xs:complexType>
<xs:complexType name="Choaza">
  <xs:complexContent>
    <xs:extension base="Gyoseiku">
      <xs:sequence>
        <xs:element name="Chiban" type="Chiban" minOccurs="0" />
        <xs:element name="Gaikufugo" type="Gaikufugo" minOccurs="0" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="Gaikufugo">
  <xs:complexContent>
    <xs:extension base="Shosaichimei">
      <xs:sequence>
        <xs:element name="Jyukyobango" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="Jyukyobango">
  <xs:complexContent>
    <xs:extension base="Shosaichimei">
      <xs:attribute name="doromei" type="CharacterString" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="Chiban">
  <xs:complexContent>
    <xs:extension base="Shosaichimei">
      <xs:attribute name="edaban" type="CharacterString"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:attributeGroup name="IM_ObjectIdentification">
  <xs:attribute name="id" type="xs:ID" />
  <xs:attribute name="uuid" type="xs:string" />
</xs:attributeGroup>
<xs:simpleType name="CharacterString">
  <xs:restriction base="xs:string" />
</xs:simpleType>
</xs:schema>
```

マッピング規則

この PI タイプの符号化規則は、ISO19118 AnnexA に準拠している。

場所型のサンプル

作成した、住所型の場所型のサンプルを、表附 2-13 に示す。

実際のデータの形式は空間参照系管理サービスに依存するが、ここでは例として XML 形

式での作成例を示す。

表附 2-13 住所型の場所型のサンプル (XML 形式)

以下、場所型のインスタンス部分の抜粋

```
<jci:PI_LocationType id="lt_add">
  <jps:SI_LocationType.theme>住所の位置の指定</jps:SI_LocationType.theme>
  <jps:SI_LocationType.definition>行政界及び海岸線によって構成される、地方自治法に定められた行政上の区域</jps:SI_LocationType.definition>
  <jps:SI_LocationType.territoryOfUse>日本全国</jps:SI_LocationType.territoryOfUse>
  <jps:SI_LocationType.owner>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>電子地図/建設情報連携小委員会</CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationType.owner>
  <jci:PI_LocationType.name>住所型</jci:PI_LocationType.name>
  <jci:PI_LocationType.identification>
    <jci:PI_Type>
      <PI_Type.applicationSchema>http://xxxxx.jp/pitype/add.html</PI_Type.applicationSchema>
      <PI_Type.encodingRule>http://xxxxx.jp/pitype/add.html</PI_Type.encodingRule>
      <PI_Type.mappingRule>http://xxxxx.jp/pitype/add.html</PI_Type.mappingRule>
    </jci:PI_Type>
  </jci:PI_LocationType.identification>
</jci:PI_LocationType>
```

2-2-2-2. 管理路線型

場所型の作成方法の検討

「管理路線型」の場所型の作成例は、表附 2-14 のようになる。また、地理識別子の記述方法を定める管理路線型の PI タイプの作成例を、別途示す。

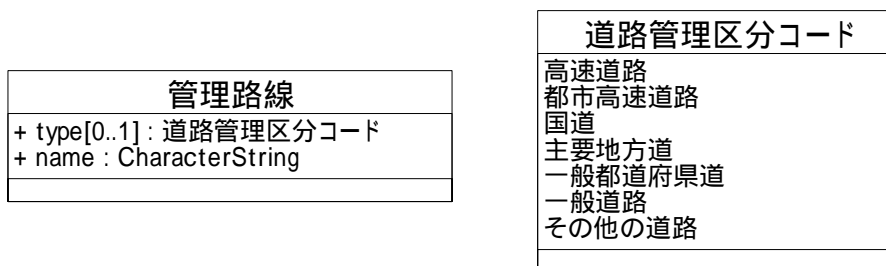
表附 2-14 「管理路線型」の場所型の作成例

属性・関連役割	値
名前	管理路線型
主題	管理路線の位置の指定
識別	(別途掲載)
定義	国土交通省の組織である国道事務所及び河川事務所が管轄する路線範囲
使用地域	日本全国
責任者	電子地図/建設情報連携小委員会

管理路線型の PI タイプ例

応用スキーマ

管理路線型の応用スキーマを、図附 2-8 及び表附 2-15 に示す。



図附 2-8 管理路線型の応用スキーマ

表附 2-15 管理路線型の応用スキーマ定義文書

型	名前	管理路線	上位となる型	なし
	定義	国土交通省の組織である国道事務所及び河川事務所が管轄する路線		
抽象/具象区分	具象			
属性	属性名	種別		
	定義	管理路線の管理区分		
	型	道路管理区分コード		
	多重度	0..1	順序指標	なし
属性	役割名	名前		
	定義	管理路線の名称		
	型	CharacterString		
	多重度	1	順序指標	なし

符号化規則

図附 2-8 に示した応用スキーマの符号化規則を、に示す。

なお、管理路線のタグ名称は“ Rd ”とする。

表附 2-16 管理路線型の符号化規則

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Rd" type="Rd" />
  <xs:complexType name="Rd">
    <xs:attribute name="type" type="RdType"/>
    <xs:attribute name="name" type="CharacterString"/>
  </xs:complexType>
  <xs:simpleType name="RdType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="高速道路"/>
      <xs:enumeration value="都市高速道路"/>
      <xs:enumeration value="国道"/>
      <xs:enumeration value="主要地方道"/>
      <xs:enumeration value="一般都道府県道"/>
      <xs:enumeration value="一般道路"/>
      <xs:enumeration value="その他の道路"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="CharacterString">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
```

マッピング規則

この PI タイプの符号化規則は、ISO19118 AnnexA に準拠している。

場所型のサンプル

作成した、管理路線型の場所型のサンプルを、表附 2-17 に示す。

実際のデータの形式は空間参照系管理サービスに依存するが、ここでは例として XML 形式での作成例を示す。

表附 2-17 管理路線型の場所型のサンプル (XML 形式)

以下、場所型のインスタンス部分の抜粋

```
<jci:PI_LocationType id="lt_rd">
  <jps:SI_LocationType.theme>管理路線の位置の指定</jps:SI_LocationType.theme>
  <jps:SI_LocationType.definition>国土交通省の組織である国道事務所及び河川事務所が管轄する路線範囲
</jps:SI_LocationType.definition>
  <jps:SI_LocationType.territoryOfUse>日本全国</jps:SI_LocationType.territoryOfUse>
  <jps:SI_LocationType.owner>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>電子地図/建設情報連携小委員会
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationType.owner>
  <jci:PI_LocationType.name>管理路線型</jci:PI_LocationType.name>
  <jci:PI_LocationType.identification>
    <jci:PI_Type>
      <PI_Type.applicationSchema>http://xxxxx.jp/pitype/rd.html</PI_Type.applicationSchema>
      <PI_Type.encodingRule>http://xxxxx.jp/pitype/rd.html</PI_Type.encodingRule>
      <PI_Type.mappingRule>http://xxxxx.jp/pitype/rd.html</PI_Type.mappingRule>
    </jci:PI_Type>
  </jci:PI_LocationType.identification>
</jci:PI_LocationType>
```


2-2-2-3. 道路距離標型

場所型の作成方法の検討

「道路距離標型」の場所型の作成例は、表附 2-18 のようになる。また、地理識別子の記述方法を定める道路距離標型の PI タイプの作成例を、別途示す。

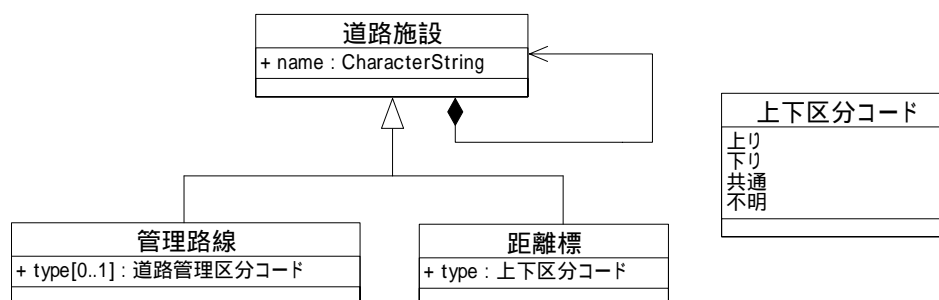
表附 2-18 「道路距離標型」の場所型の作成例

属性・関連役割	値
名前	道路距離標型
主題	距離標による道路延長方向の位置の指定
識別	(別途掲載)
定義	道路起点からの延長を表示した標識で、道路端に設置してある杭や鉄
使用地域	日本全国
責任者	電子地図/建設情報連携小委員会

道路距離標型の PI タイプ例

応用スキーマ

道路距離標型の応用スキーマを、図附 2-9 及び表附 2-19 に示す。



図附 2-9 道路距離標型の応用スキーマ

表附 2-19 道路距離標型の応用スキーマ定義文書

型	名前	道路施設	上位となる型	なし	
	定義	道路起点からの延長を表示した標識で、道路端に設置してある杭や鉄			
	抽象/具象区分	具象			
	属性	属性名	名前		
		定義	道路施設を識別する値		
		型	CharacterString		
		多重度	0..1	順序指標	なし
	役割	役割名	要素		
		定義	道路施設を構成する要素		
		型	道路施設		
多重度		1	順序指標	なし	
型	名前	管理路線	上位となる型	道路施設	
	定義	国土交通省の組織である国道事務所及び河川事務所が管轄する路線			
	抽象/具象区分	具象			
	属性	属性名	種別		
		定義	管理路線の管理区分		
		型	道路管理区分コード		
		多重度	0..1	順序指標	なし
	役割	役割名	距離標		
		定義	道路の起点から終点までに置かれた杭や鉄		
		型	距離標		
多重度		0..1	順序指標	なし	
型	名前	距離標	上位となる型	道路施設	
	定義	道路起点からの延長を表示した標識で、道路端に設置してある杭や鉄			
	抽象/具象区分	具象			
	属性	属性名	上下の別		
		定義	距離表の上下区分の別		
		型	上下区分コード		
		多重度	1	順序指標	なし

符号化規則

図附 2-9 に示した応用スキーマの符号化規則を、表附 2-20 に示す。

なお、距離標のタグ名称は“Kp”とする。

表附 2-20 道路距離標型の符号化規則

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="RdF" type="RdF"/>
  <xs:element name="Rd" type="Rd" substitutionGroup="RdF"/>
  <xs:element name="Kp" type="Kp" substitutionGroup="RdF"/>
  <xs:complexType name="RdF">
    <xs:sequence>
      <xs:element ref="RdF" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="name" type="CharacterString"/>
  </xs:complexType>
  <xs:complexType name="Rd">
    <xs:complexContent>
      <xs:extension base="RdF">
        <xs:sequence>
          <xs:element ref="Kp" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="type" type="RdType"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="Kp">
    <xs:complexContent>
      <xs:extension base="RdF">
        <xs:attribute name="type" type="KpType"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:simpleType name="RdType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="高速道路"/>
      <xs:enumeration value="都市高速道路"/>
      <xs:enumeration value="国道"/>
      <xs:enumeration value="主要地方道"/>
      <xs:enumeration value="一般都道府県道"/>
      <xs:enumeration value="一般道路"/>
      <xs:enumeration value="その他の道路"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="KpType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="上"/>
      <xs:enumeration value="下"/>
      <xs:enumeration value="共通"/>
      <xs:enumeration value="不明"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="CharacterString">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>

```

マッピング規則

この PI タイプの符号化規則は、ISO19118 AnnexA に準拠している。

場所型のサンプル

作成した、道路距離標型の場所型のサンプルを、表附 2-21 に示す。

実際のデータの形式は空間参照系管理サービスに依存するが、ここでは例として XML 形式での作成例を示す。

表附 2-21 道路距離標型の場所型のサンプル (XML 形式)

以下、場所型のインスタンス部分の抜粋

```
<jci:PI_LocationType id="lt_rdkp">
  <jps:SI_LocationType.theme>距離標による道路延長方向の位置の指定</jps:SI_LocationType.theme>
  <jps:SI_LocationType.definition>道路起点からの延長を表示した標識で、道路端に設置してある杭や鋸
  </jps:SI_LocationType.definition>
  <jps:SI_LocationType.territoryOfUse>日本全国</jps:SI_LocationType.territoryOfUse>
  <jps:SI_LocationType.owner>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>電子地図/建設情報連携小委員会
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationType.owner>
  <jci:PI_LocationType.name>道路距離標型</jci:PI_LocationType.name>
  <jci:PI_LocationType.identification>
    <jci:PI_Type>
      <PI_Type.applicationSchema>http://xxxxx.jp/pitype/rdkp.html</PI_Type.applicationSchema>
      <PI_Type.encodingRule>http://xxxxx.jp/pitype/rdkp.html</PI_Type.encodingRule>
      <PI_Type.mappingRule>http://xxxxx.jp/pitype/rdkp.html</PI_Type.mappingRule>
    </jci:PI_Type>
  </jci:PI_LocationType.identification>
</jci:PI_LocationType>
```

2-2-2-4. 交差点型

場所型の作成方法の検討

「交差点型」の場所型の作成例は、表附 2-22 のようになる。また、地理識別子の記述方法を定める交差点型の PI タイプの作成例を、別途示す。

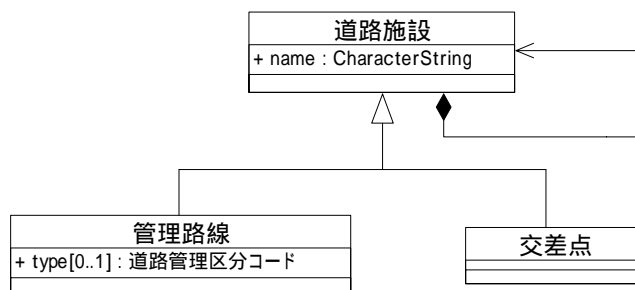
表附 2-22 「交差点型」の場所型の作成例

属性・関連役割	値
名前	交差点型
主題	交差点の位置の指定
識別	(別途掲載)
定義	十字路、T 字路など、2 以上の車道が交わる部分
使用地域	日本全国
責任者	電子地図/建設情報連携小委員会

交差点型の PI タイプ例

応用スキーマ

交差点型の応用スキーマを、図附 2-10 及び表附 2-23 に示す。



図附 2-10 交差点型の応用スキーマ

表附 2-23 交差点型の応用スキーマ定義文書

型	名前	道路施設	上位となる型	なし	
	定義	人や車が通行するための施設			
	抽象/具象区分	具象			
	属性	属性名	名前		
		定義	道路施設を識別する値		
		型	CharacterString		
		多重度	0..1	順序指標	なし
	役割	役割名	要素		
		定義	道路施設を構成する要素		
		型	道路施設		
多重度		1	順序指標	なし	
型	名前	管理路線	上位となる型	道路施設	
	定義	国土交通省の組織である国道事務所及び河川事務所が管轄する路線			
	抽象/具象区分	具象			
	属性	属性名	種別		
		定義	管理路線の管理区分		
		型	道路管理区分コード		
		多重度	0..1	順序指標	なし
	役割	役割名	交差点		
		定義	十字路、T字路など、2以上の車道が交わる部分		
		型	交差点		
多重度		0..1	順序指標	なし	
型	名前	交差点	上位となる型	道路施設	
	定義	十字路、T字路など、2以上の車道が交わる部分			
	抽象/具象区分	具象			

符号化規則

図附 2-10 に示した応用スキーマの符号化規則を、表附 2-24 に示す。

なお、交差点のタグ名称は “Its ” とする。

表附 2-24 交差点型の符号化規則

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="RdF" type="RdF"/>
  <xs:element name="Rd" type="Rd" substitutionGroup="RdF"/>
  <xs:element name="Kp" type="Kp" substitutionGroup="RdF"/>
  <xs:complexType name="RdF">
    <xs:sequence>
      <xs:element ref="RdF" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="name" type="CharacterString"/>
  </xs:complexType>
  <xs:complexType name="Rd">
    <xs:complexContent>
      <xs:extension base="RdF">
        <xs:sequence>
          <xs:element ref="It" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="type" type="RdType"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="Its">
    <xs:complexContent>
      <xs:extension base="RdF"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:simpleType name="RdType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="高速道路"/>
      <xs:enumeration value="都市高速道路"/>
      <xs:enumeration value="国道"/>
      <xs:enumeration value="主要地方道"/>
      <xs:enumeration value="一般都道府県道"/>
      <xs:enumeration value="一般道路"/>
      <xs:enumeration value="その他の道路"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="CharacterString">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>

```

マッピング規則

この PI タイプの符号化規則は、ISO19118 AnnexA に準拠している。

場所型のサンプル

作成した、交差点型の場所型のサンプルを、表附 2-25 に示す。

実際のデータの形式は空間参照系管理サービスに依存するが、ここでは例として XML 形

式での作成例を示す。

表附 2-25 交差点型の場所型のサンプル (XML 形式)

以下、場所型のインスタンス部分の抜粋

```
<jci:PI_LocationType id="lt_its">
  <jps:SI_LocationType.theme>交差点の位置の指定</jps:SI_LocationType.theme>
  <jps:SI_LocationType.definition>道路起点からの延長を表示した標識で、道路端に設置してある杭や鉄
  </jps:SI_LocationType.definition>
  <jps:SI_LocationType.territoryOfUse>日本全国</jps:SI_LocationType.territoryOfUse>
  <jps:SI_LocationType.owner>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>電子地図/建設情報連携小委員会
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationType.owner>
  <jci:PI_LocationType.name>交差点型</jci:PI_LocationType.name>
  <jci:PI_LocationType.identification>
    <jci:PI_Type>
      <PI_Type.applicationSchema>http://xxxxx.jp/pitype/its.html</PI_Type.applicationSchema>
      <PI_Type.encodingRule>http://xxxxx.jp/pitype/its.html</PI_Type.encodingRule>
      <PI_Type.mappingRule>http://xxxxx.jp/pitype/its.html</PI_Type.mappingRule>
    </jci:PI_Type>
  </jci:PI_LocationType.identification>
</jci:PI_LocationType>
```


2-2-2-5. 管理河川型

場所型の作成方法の検討

「管理河川型」の場所型の作成例は、表附 2-26 のようになる。また、地理識別子の記述方法を定める管理河川型の PI タイプの作成例を、別途示す。

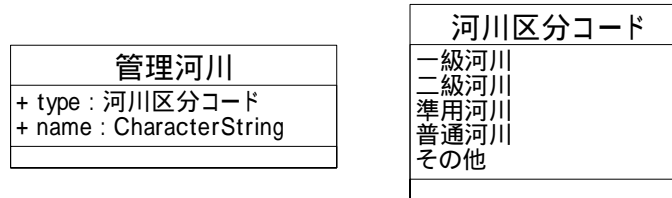
表附 2-26 「管理河川型」の場所型の作成例

属性・関連役割	値
名前	管理河川型
主題	管理河川の位置の指定
識別	(別途掲載)
定義	国土交通省の組織である国道事務所及び河川事務所が管轄する河川範囲
使用地域	日本全国
責任者	電子地図/建設情報連携小委員会

管理河川型の PI タイプ例

応用スキーマ

管理河川型の応用スキーマを、図附 2-11 及び表附 2-27 に示す。



図附 2-11 管理河川型の応用スキーマ

表附 2-27 管理河川型の応用スキーマ定義文書

型	名前	管理河川	上位となる型	なし
	定義	国土交通省の組織である国道事務所及び河川事務所が管轄する河川		
	抽象/具象区分	具象		
属性	属性名	種別		
	定義	管理河川の区分		
	型	河川区分コード		
	多重度	0..1	順序指標	なし
属性	役割名	名前		
	定義	管理河川の名称		
	型	CharacterString		
	多重度	1	順序指標	なし

符号化規則

図附 2-11 に示した応用スキーマの符号化規則を、表附 2-28 に示す。

なお、管理河川のタグ名称は“ Riv ”とする。

表附 2-28 管理河川型の符号化規則

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Riv" type="Riv"/>
  <xs:complexType name="Riv">
    <xs:attribute name="type" type="RivType"/>
    <xs:attribute name="name" type="CharacterString"/>
  </xs:complexType>
  <xs:simpleType name="RivType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="一級河川直轄"/>
      <xs:enumeration value="一級河川指定"/>
      <xs:enumeration value="二級河川"/>
      <xs:enumeration value="準用河川"/>
      <xs:enumeration value="普通河川"/>
      <xs:enumeration value="その他"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="CharacterString">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
```

マッピング規則

この PI タイプの符号化規則は、ISO19118 AnnexA に準拠している。

場所型のサンプル

作成した、管理河川型の場所型のサンプルを、表附 2-29 に示す。

実際のデータの形式は空間参照系管理サービスに依存するが、ここでは例として XML 形式での作成例を示す。

表附 2-29 管理河川型の場所型のサンプル (XML 形式)

以下、場所型のインスタンス部分の抜粋

```
<jci:PI_LocationType id="lt_riv">
  <jps:SI_LocationType.theme>管理河川の位置の指定</jps:SI_LocationType.theme>
  <jps:SI_LocationType.definition>国土交通省の組織である国道事務所及び河川事務所が管轄する河川範囲
</jps:SI_LocationType.definition>
  <jps:SI_LocationType.territoryOfUse>日本全国</jps:SI_LocationType.territoryOfUse>
  <jps:SI_LocationType.owner>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>電子地図/建設情報連携小委員会
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationType.owner>
  <jci:PI_LocationType.name>管理河川型</jci:PI_LocationType.name>
  <jci:PI_LocationType.identification>
    <jci:PI_Type>
      <PI_Type.applicationSchema>http://xxxxx.jp/pitype/riv.html</PI_Type.applicationSchema>
      <PI_Type.encodingRule>http://xxxxx.jp/pitype/riv.html</PI_Type.encodingRule>
      <PI_Type.mappingRule>http://xxxxx.jp/pitype/riv.html</PI_Type.mappingRule>
    </jci:PI_Type>
  </jci:PI_LocationType.identification>
</jci:PI_LocationType>
```

2-2-2-6. 河川距離標型

場所型の作成方法の検討

「河川距離標型」の場所型の作成例は、表附 2-30 のようになる。また、地理識別子の記述方法を定める河川距離標型の PI タイプの作成例を、別途示す。

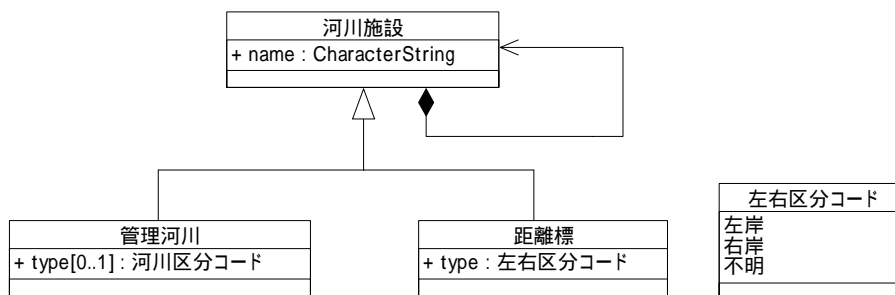
表附 2-30 「河川距離標型」の場所型の作成例

属性・関連役割	値
名前	河川距離標型
主題	距離標による河川延長方向の位置の指定
識別	(別途掲載)
定義	河川起点からの延長を表示した標識で、河川端に設置してある杭や鉄
使用地域	日本全国
責任者	電子地図/建設情報連携小委員会

河川距離標型の PI タイプ例

応用スキーマ

河川距離標型の応用スキーマを、図附 2-12 及び表附 2-31 に示す。



図附 2-12 河川距離標型の応用スキーマ

表附 2-31 河川距離標型の応用スキーマ定義文書

型	名前	河川施設	上位となる型	なし	
	定義	河川事務所が管理する河川に関する施設			
	抽象/具象区分	具象			
	属性	属性名	名前		
		定義	河川施設を識別する値		
		型	CharacterString		
		多重度	0..1	順序指標	なし
	役割	役割名	要素		
		定義	河川施設を構成する要素		
		型	河川施設		
多重度		1	順序指標	なし	
型	名前	管理河川	上位となる型	河川施設	
	定義	国土交通省の組織である国道事務所及び河川事務所が管轄する河川			
	抽象/具象区分	具象			
	属性	属性名	種別		
		定義	管理河川の区分		
		型	河川区分コード		
		多重度	0..1	順序指標	なし
	役割	役割名	距離標		
		定義	河川の起点から終点までに置かれた杭や鉄		
		型	距離標		
多重度		0..1	順序指標	なし	
型	名前	距離標	上位となる型	河川施設	
	定義	河川の起点からの延長を表示した標識で、河川端に設置してある杭や鉄			
	抽象/具象区分	具象			
	属性	属性名	左右の別		
		定義	距離標の左岸右岸の別		
		型	左右区分コード		
		多重度	1	順序指標	なし

符号化規則

図附 2-12 に示した応用スキーマの符号化規則を、表附 2-32 に示す。

なお、距離標のタグ名称は“Kp”とする。

表附 2-32 河川距離標型の符号化規則

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="RivF" type="RivF"/>
  <xs:element name="Riv" type="Riv" substitutionGroup="RivF"/>
  <xs:element name="Kp" type="Kp" substitutionGroup="RivF"/>
  <xs:complexType name="RivF">
    <xs:sequence>
      <xs:element ref="RivF" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="name" type="CharacterString"/>
  </xs:complexType>
  <xs:complexType name="Riv">
    <xs:complexContent>
      <xs:extension base="RivF">
        <xs:sequence>
          <xs:element ref="Kp" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="type" type="RivType"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="Kp">
    <xs:complexContent>
      <xs:extension base="RivF">
        <xs:attribute name="type" type="RivKpType"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:simpleType name="RivType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="一級河川直轄"/>
      <xs:enumeration value="一級河川指定"/>
      <xs:enumeration value="二級河川"/>
      <xs:enumeration value="準用河川"/>
      <xs:enumeration value="普通河川"/>
      <xs:enumeration value="その他"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="RivKpType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="左岸"/>
      <xs:enumeration value="右岸"/>
      <xs:enumeration value="不明"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="CharacterString">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>

```

マッピング規則

この PI タイプの符号化規則は、ISO19118 AnnexA に準拠している。

場所型のサンプル

作成した、河川距離標型の場所型のサンプルを、表附 2-33 に示す。

実際のデータの形式は空間参照系管理サービスに依存するが、ここでは例として XML 形式での作成例を示す。

表附 2-33 河川距離標型の場所型のサンプル (XML 形式)

以下、場所型のインスタンス部分の抜粋

```
<jci:PI_LocationType id="lt_Rivkp">
  <jps:SI_LocationType.theme>距離標による河川延長方向の位置の指定</jps:SI_LocationType.theme>
  <jps:SI_LocationType.definition>河川起点からの延長を表示した標識で、河川端に設置してある杭や鉄
  </jps:SI_LocationType.definition>
  <jps:SI_LocationType.territoryOfUse>日本全国</jps:SI_LocationType.territoryOfUse>
  <jps:SI_LocationType.owner>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>電子地図/建設情報連携小委員会
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationType.owner>
  <jci:PI_LocationType.name>河川距離標型</jci:PI_LocationType.name>
  <jci:PI_LocationType.identification>
    <jci:PI_Type>
      <PI_Type.applicationSchema>http://xxxxx.jp/pitype/rivkp.html</PI_Type.applicationSchema>
      <PI_Type.encodingRule>http://xxxxx.jp/pitype/rivkp.html</PI_Type.encodingRule>
      <PI_Type.mappingRule>http://xxxxx.jp/pitype/rivkp.html</PI_Type.mappingRule>
    </jci:PI_Type>
  </jci:PI_LocationType.identification>
</jci:PI_LocationType>
```

2-2-2-7. 水系型

場所型の作成方法の検討

「水系型」の場所型の作成例は、表附 2-34 のようになる。また、地理識別子の記述方法を定める水系型の PI タイプの作成例を、別途示す。

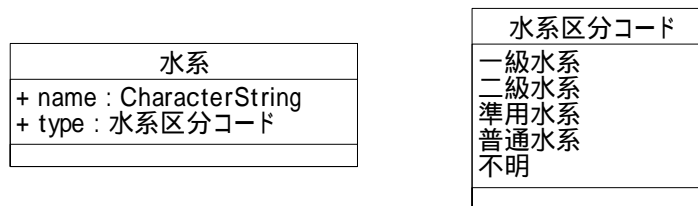
表附 2-34 「水系型」の場所型の作成例

属性・関連役割	値
名前	水系型
主題	水系による範囲の指定
識別	(別途掲載)
定義	本川と本川に流れ込む支川を一連のまとまりとした範囲
使用地域	日本全国
責任者	電子地図/建設情報連携小委員会

水系型の PI タイプ例

応用スキーマ

水系型の応用スキーマを、図附 2-13 及び表附 2-35 に示す。



図附 2-13 水系型の応用スキーマ

表附 2-35 水系型の応用スキーマ定義文書

型	名前	水系	上位となる型	なし
	定義	本川と本川に流れ込む支川を一連のまとまりとした範囲		
	抽象/具象区分	具象		
属性	属性名	種別		
	定義	水系の区分		
	型	水系区分コード		
	多重度	1	順序指標	なし
属性	役割名	名前		
	定義	水系の名称		
	型	CharacterString		
	多重度	1	順序指標	なし

符号化規則

図附 2-13 に示した応用スキーマの符号化規則を、表附 2-36 に示す。

なお、水系のタグ名称は“ RivRa ”とする。

表附 2-36 水系型の符号化規則

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="RivRa" type="RivRa" />
  <xs:complexType name="RivRa">
    <xs:attribute name="type" type="RaType"/>
    <xs:attribute name="name" type="CharacterString"/>
  </xs:complexType>
  <xs:simpleType name="RaType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="一級水系" />
      <xs:enumeration value="二級水系" />
      <xs:enumeration value="準用水系" />
      <xs:enumeration value="普通水系" />
      <xs:enumeration value="不明" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="CharacterString">
    <xs:restriction base="xs:string" />
  </xs:simpleType>
</xs:schema>
```

マッピング規則

この PI タイプの符号化規則は、ISO19118 AnnexA に準拠している。

場所型のサンプル

作成した、水系型の場所型のサンプルを、表附 2-37 に示す。

実際のデータの形式は空間参照系管理サービスに依存するが、ここでは例として XML 形式での作成例を示す。

表附 2-37 水系型の場所型のサンプル (XML 形式)

以下、場所型のインスタンス部分の抜粋

```
<jci:PI_LocationType id="lt_rivra">
  <jps:SI_LocationType.theme>水系の範囲の指定</jps:SI_LocationType.theme>
  <jps:SI_LocationType.definition>本川と本川に流れ込む支川を一連のまとりとした範囲
</jps:SI_LocationType.definition>
  <jps:SI_LocationType.territoryOfUse>日本全国</jps:SI_LocationType.territoryOfUse>
  <jps:SI_LocationType.owner>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>電子地図/建設情報連携小委員会
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationType.owner>
  <jci:PI_LocationType.name>水系型</jci:PI_LocationType.name>
  <jci:PI_LocationType.identification>
    <jci:PI_Type>
      <PI_Type.applicationSchema>http://xxxxx.jp/pitype/rivra.html</PI_Type.applicationSchema>
      <PI_Type.encodingRule>http://xxxxx.jp/pitype/rivra.html</PI_Type.encodingRule>
      <PI_Type.mappingRule>http://xxxxx.jp/pitype/rivra.html</PI_Type.mappingRule>
    </jci:PI_Type>
  </jci:PI_LocationType.identification>
</jci:PI_LocationType>
```

2-2-2-8. 流域型

場所型の作成方法の検討

「流域型」の場所型の作成例は、表附 2-38 のようになる。また、地理識別子の記述方法を定める流域型の PI タイプの作成例を、別途示す。

表附 2-38 「流域型」の場所型の作成例

属性・関連役割	値
名前	流域型
主題	流域による範囲の指定
識別	(別途掲載)
定義	分水嶺を境界とする川の自然集水域
使用地域	日本全国
責任者	電子地図/建設情報連携小委員会

流域型の PI タイプ例

応用スキーマ

流域型の応用スキーマを、図附 2-14 及び表附 2-39 に示す。

流域
+ name : CharacterString

図附 2-14 流域型の応用スキーマ

表附 2-39 流域型の応用スキーマ定義文書

型	名前	流域	上位となる型	なし
	定義	分水嶺を境界とする川の自然集水域		
	抽象/具象区分	具象		
属性	役割名	名前		
	定義	流域の名称		
	型	CharacterString		
	多重度	1	順序指標	なし

符号化規則

図附 2-14 に示した応用スキーマの符号化規則を、表附 2-40 に示す。

なお、流域のタグ名称は “ RivWs ” とする。

表附 2-40 流域型の符号化規則

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="RivWs" type="RivWs"/>
  <xs:complexType name="RivWs">
    <xs:attribute name="name" type="CharacterString"/>
  </xs:complexType>
  <xs:simpleType name="CharacterString">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
```

マッピング規則

この PI タイプの符号化規則は、ISO19118 AnnexA に準拠している。

場所型のサンプル

作成した、流域型の場所型のサンプルを、表附 2-41 に示す。

実際のデータの形式は空間参照系管理サービスに依存するが、ここでは例として XML 形式での作成例を示す。

表附 2-41 流域型の場所型のサンプル (XML 形式)

以下、場所型のインスタンス部分の抜粋

```
<jci:PI_LocationType id="lt_rivws">
  <jps:SI_LocationType.theme>流域の範囲の指定</jps:SI_LocationType.theme>
  <jps:SI_LocationType.definition>分水嶺を境界とする川の自然集水域</jps:SI_LocationType.definition>
  <jps:SI_LocationType.territoryOfUse>日本全国</jps:SI_LocationType.territoryOfUse>
  <jps:SI_LocationType.owner>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>電子地図/建設情報連携小委員会
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationType.owner>
  <jci:PI_LocationType.name>流域型</jci:PI_LocationType.name>
  <jci:PI_LocationType.identification>
    <jci:PI_Type>
      <PI_Type.applicationSchema>http://xxxxx.jp/pitype/rivws.html</PI_Type.applicationSchema>
      <PI_Type.encodingRule>http://xxxxx.jp/pitype/rivws.html</PI_Type.encodingRule>
      <PI_Type.mappingRule>http://xxxxx.jp/pitype/rivws.html</PI_Type.mappingRule>
    </jci:PI_Type>
  </jci:PI_LocationType.identification>
</jci:PI_LocationType>
```

2-2-2-9. 測点型

場所型の作成方法の検討

「測点型」の場所型の作成例は、表附 2-42 のようになる。また、地理識別子の記述方法を定める測点型の PI タイプの作成例を、別途示す。

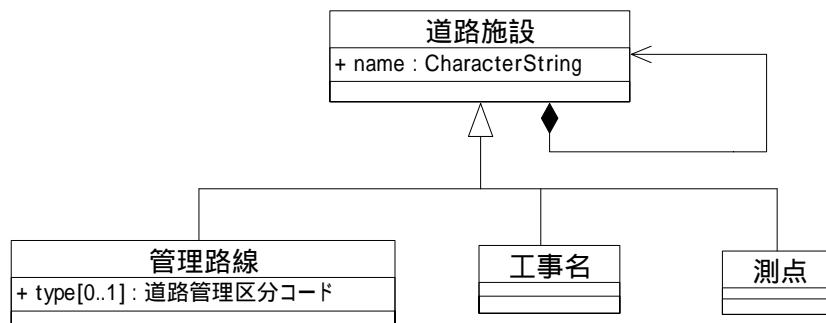
表附 2-42 「測点型」の場所型の作成例

属性・関連役割	値
名前	測点型
主題	測点による道路延長方向の位置の指定
識別	(別途掲載)
定義	路線測量によって作成された計測点
使用地域	日本全国
責任者	電子地図/建設情報連携小委員会

測点型の PI タイプ例

応用スキーマ

測点型の応用スキーマを、図附 2-15 及び表附 2-43 に示す。



図附 2-15 測点型の応用スキーマ

表附 2-43 測点型の応用スキーマ定義文書

型	名前	道路施設	上位となる型	なし	
	定義	人や車が通行するための施設			
	抽象/具象区分	具象			
	属性	属性名	名前		
		定義	道路施設を識別する値		
		型	CharacterString		
		多重度	1	順序指標	なし
	役割	役割名	要素		
		定義	道路施設を構成する要素		
		型	道路施設		
多重度		1	順序指標	なし	
型	名前	管理路線	上位となる型	道路施設	
	定義	国土交通省の組織である国道事務所及び河川事務所が管轄する路線			
	抽象/具象区分	具象			
	属性	属性名	種別		
		定義	管理路線の管理区分		
		型	道路管理区分コード		
		多重度	0..1	順序指標	なし
	役割	役割名	工事名		
		定義	行われた道路工事を識別する値		
		型	工事名		
多重度		0..1	順序指標	なし	
型	名前	工事名	上位となる型	道路施設	
	定義	行われた道路工事を識別する値			
	抽象/具象区分	具象			
	属性	属性名	名前		
		定義	行われた道路工事を識別する値		
		型	CharacterString		
		多重度	1	順序指標	なし
	役割	役割名	測点		
		定義	路線測量によって作成された計測点		
		型	測点		
多重度		0..1	順序指標	なし	
型	名前	測点	上位となる型	道路施設	
	定義	路線測量によって作成された計測点			
	抽象/具象区分	具象			
	属性	属性名	名前		
		定義	路線測量によって作成された計測点を識別する値		
		型	CharacterString		
		多重度	1	順序指標	なし

符号化規則

図附 2-15 に示した応用スキーマの符号化規則を、表附 2-44 に示す。

なお、工事名のタグ名称は“ Rc ”とし、測点のタグ名称は“ Sp ”とする。

表附 2-44 測点型の符号化規則

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="RdF" type="RdF"/>
  <xs:element name="Rd" type="Rd" substitutionGroup="RdF"/>
  <xs:element name="Rc" type="Rc" substitutionGroup="RdF"/>
  <xs:element name="Sp" type="Sp" substitutionGroup="RdF"/>
  <xs:complexType name="RdF">
    <xs:sequence>
      <xs:element ref="RdF" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="name" type="CharacterString"/>
  </xs:complexType>
  <xs:complexType name="Rd">
    <xs:complexContent>
      <xs:extension base="RdF">
        <xs:sequence>
          <xs:element ref="Rc" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="type" type="RdType"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="Rc">
    <xs:complexContent>
      <xs:extension base="RdF">
        <xs:sequence>
          <xs:element ref="Sp" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="Sp">
    <xs:complexContent>
      <xs:extension base="RdF"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:simpleType name="RdType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="高速道路"/>
      <xs:enumeration value="都市高速道路"/>
      <xs:enumeration value="国道"/>
      <xs:enumeration value="主要地方道"/>
      <xs:enumeration value="一般都道府県道"/>
      <xs:enumeration value="一般道路"/>
      <xs:enumeration value="その他の道路"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```



```
<xs:simpleType name="CharacterString">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
</xs:schema>
```

マッピング規則

この PI タイプの符号化規則は、ISO19118 AnnexA に準拠している。

場所型のサンプル

作成した、測点型の場所型のサンプルを、表附 2-45 に示す。

実際のデータの形式は空間参照系管理サービスに依存するが、ここでは例として XML 形式での作成例を示す。

表附 2-45 測点型の場所型のサンプル (XML 形式)

以下、場所型のインスタンス部分の抜粋

```
<jci:PI_LocationType id="lt_sp">
  <jps:SI_LocationType.theme>測点による道路延長方向の位置の指定</jps:SI_LocationType.theme>
  <jps:SI_LocationType.definition>路線測量によって作成された計測点</jps:SI_LocationType.definition>
  <jps:SI_LocationType.territoryOfUse>日本全国</jps:SI_LocationType.territoryOfUse>
  <jps:SI_LocationType.owner>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>電子地図/建設情報連携小委員会
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationType.owner>
  <jci:PI_LocationType.name>測点型</jci:PI_LocationType.name>
  <jci:PI_LocationType.identification>
    <jci:PI_Type>
      <PI_Type.applicationSchema>http://xxxxx.jp/pitype/sp.html</PI_Type.applicationSchema>
      <PI_Type.encodingRule>http://xxxxx.jp/pitype/sp.html</PI_Type.encodingRule>
      <PI_Type.mappingRule>http://xxxxx.jp/pitype/sp.html</PI_Type.mappingRule>
    </jci:PI_Type>
  </jci:PI_LocationType.identification>
</jci:PI_LocationType>
```

2-2-2-10. 事務所管轄区域型

事務所管轄区域型の作成方法の検討

「事務所管轄区域型」の場所型の作成例は、表附 2-46 のようになる。また、地理識別子の記述方法を定める事務所管轄区域型の PI タイプの作成例を、別途示す。

表附 2-46 「事務所管轄区域型」の場所型の作成例

属性・関連役割	値
名前	事務所管轄区域型
主題	河川事務所及び国道事務所の範囲の指定
識別	(別途掲載)
定義	国土交通省の組織である河川及び国道事務所が管轄する河川、路線の範囲
使用地域	日本全国
責任者	電子地図/建設情報連携小委員会

事務所管轄区域型の PI タイプ例

応用スキーマ

事務所管轄区域型の応用スキーマを、図附 2-16 及び表附 2-47 事務所管轄区域型の応用ス

事務所管轄区域
+ name : CharacterString

キーマ定義文書に示す。

図附 2-16 事務所管轄区域型の応用スキーマ

表附 2-47 事務所管轄区域型の応用スキーマ定義文書

型	名前	事務所管轄区域	上位となる型	なし
	定義	国土交通省の組織である河川及び国道事務所が管轄する河川、路線の範囲		
	抽象/具象区分	具象		
属性	役割名	名前		
	定義	事務所管轄区域の名称		
	型	CharacterString		
	多重度	1	順序指標	なし

符号化規則

図附 2-16 に示した応用スキーマの符号化規則を、表附 2-48 に示す。

なお、事務所管轄区域のタグ名称は“ OfD ”とする。

表附 2-48 事務所管轄区域型の符号化規則

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="OfD" type="OfD"/>
  <xs:complexType name="OfD">
    <xs:attribute name="name" type="CharacterString"/>
  </xs:complexType>
  <xs:simpleType name="CharacterString">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
```

マッピング規則

この PI タイプの符号化規則は、ISO19118 AnnexA に準拠している。

場所型のサンプル

作成した、事務所管轄区域型の場所型のサンプルを、表附 2-49 に示す。

実際のデータの形式は空間参照系管理サービスに依存するが、ここでは例として XML 形式での作成例を示す。

表附 2-49 事務所管轄区域型の場所型のサンプル (XML 形式)

以下、場所型のインスタンス部分の抜粋

```
<jci:PI_LocationType id="lt_ofd">
  <jps:SI_LocationType.theme>河川事務所及び国道事務所の範囲の指定の範囲の
  </jps:SI_LocationType.theme>
  <jps:SI_LocationType.definition>国土交通省の組織である河川及び国道事務所が管轄する河川、路線の範囲
  </jps:SI_LocationType.definition>
  <jps:SI_LocationType.territoryOfUse>日本全国</jps:SI_LocationType.territoryOfUse>
  <jps:SI_LocationType.owner>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>電子地図/建設情報連携小委員会
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationType.owner>
  <jci:PI_LocationType.name>事務所管轄区域型</jci:PI_LocationType.name>
  <jci:PI_LocationType.identification>
    <jci:PI_Type>
      <PI_Type.applicationSchema>http://xxxxx.jp/pitype/ofd.html</PI_Type.applicationSchema>
      <PI_Type.encodingRule>http://xxxxx.jp/pitype/ofd.html</PI_Type.encodingRule>
      <PI_Type.mappingRule>http://xxxxx.jp/pitype/ofd.html</PI_Type.mappingRule>
    </jci:PI_Type>
  </jci:PI_LocationType.identification>
</jci:PI_LocationType>
```

2-2-2-11. 出張所管轄区間型

出張所管轄区間型の作成方法の検討

「出張所管轄区間型」の場所型の作成例は、表附 2-50 のようになる。また、地理識別子の記述方法を定める出張所管轄区間型の PI タイプの作成例を、別途示す。

表附 2-50 「出張所管轄区間型」の場所型の作成例

属性・関連役割	値
名前	出張所管轄区間型
主題	河川及び国道の出張所の範囲の指定
識別	(別途掲載)
定義	国土交通省の組織で事務所の出先機関である出張所が管轄する河川、路線の範囲
使用地域	日本全国
責任者	電子地図/建設情報連携小委員会

事務所管轄区域型の PI タイプ例

応用スキーマ

事務所管轄区域型の応用スキーマを、図附 2-17 及び表附 2-51 に示す。

出張所管轄区間
+ name : CharacterString

図附 2-17 出張所管轄区間型の応用スキーマ

表附 2-51 出張所管轄区間型の応用スキーマ定義文書

型	名前	出張所管轄区間	上位となる型	なし
	定義	国土交通省の組織で事務所の出先機関である出張所が管轄する河川、路線の範囲		
	抽象/具象区分	具象		
属性	役割名	名前		
	定義	出張所管轄区間の名称		
	型	CharacterString		
	多重度	1	順序指標	なし

符号化規則

図附 2-8 に示した応用スキーマの符号化規則を、表附 2-52 に示す。

なお、出張所管轄区間のタグ名称は“ BofS ”とする。

表附 2-52 出張所管轄区間型の符号化規則

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="BofS" type="BofS"/>
  <xs:complexType name="BofS">
    <xs:attribute name="name" type="CharacterString"/>
  </xs:complexType>
  <xs:simpleType name="CharacterString">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
```

マッピング規則

この PI タイプの符号化規則は、ISO19118 AnnexA に準拠している。

場所型のサンプル

作成した、出張所管轄区間型の場所型のサンプルを、表附 2-53 に示す。

実際のデータの形式は空間参照系管理サービスに依存するが、ここでは例として XML 形式での作成例を示す。

表附 2-53 出張所管轄区間型の場所型のサンプル (XML 形式)

以下、場所型のインスタンス部分の抜粋

```
<jci:PI_LocationType id="lt_ofd">
  <jps:SI_LocationType.theme>河川及び国道の出張所の範囲の指定</jps:SI_LocationType.theme>
  <jps:SI_LocationType.definition>国土交通省の組織で事務所の出先機関である出張所が管轄する河川、路線の
    範囲</jps:SI_LocationType.definition>
  <jps:SI_LocationType.territoryOfUse>日本全国</jps:SI_LocationType.territoryOfUse>
  <jps:SI_LocationType.owner>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>電子地図/建設情報連携小委員会
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationType.owner>
  <jci:PI_LocationType.name>出張所管轄区間型</jci:PI_LocationType.name>
  <jci:PI_LocationType.identification>
    <jci:PI_Type>
      <PI_Type.applicationSchema>http://xxxxx.jp/pitype/bofs.html</PI_Type.applicationSchema>
      <PI_Type.encodingRule>http://xxxxx.jp/pitype/bofs.html</PI_Type.encodingRule>
      <PI_Type.mappingRule>http://xxxxx.jp/pitype/bofs.html</PI_Type.mappingRule>
    </jci:PI_Type>
  </jci:PI_LocationType.identification>
</jci:PI_LocationType>
```

2-2-2-12. 施設型

施設型の作成方法の検討

「施設型」の場所型の作成例は、表附 2-54 のようになる。また、地理識別子の記述方法を定める施設型の PI タイプの作成例を、別途示す。

表附 2-54 「施設型」の場所型の作成例

属性・関連役割	値
名前	施設型
主題	施設の位置の指定
識別	(別途掲載)
定義	河川及び国道事務所が管理する河川管理施設と道路管理施設、並びに公共と民間の施設
使用地域	日本全国
責任者	電子地図/建設情報連携小委員会

施設型の PI タイプ例

応用スキーマ

施設型の応用スキーマを、図附 2-18 及び表附 2-55 に示す。

施設
+ name : CharacterString

図附 2-18 施設型の応用スキーマ

表附 2-55 施設型の応用スキーマ定義文書

型	名前	施設	上位となる型	なし
属性	定義	河川及び国道事務所が管理する河川管理施設と道路管理施設、並びに公共と民間の施設		
	抽象/具象区分	具象		
	役割名	名前		
	定義	施設の名称		
	型	CharacterString		
	多重度	1	順序指標	なし

符号化規則

図附 2-8 に示した応用スキーマの符号化規則を、表附 2-56 に示す。

なお、施設のタグ名称は“Fac”とする。

表附 2-56 施設型の符号化規則

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Fac" type="Fac"/>
  <xs:complexType name="Fac">
    <xs:attribute name="name" type="CharacterString"/>
  </xs:complexType>
  <xs:simpleType name="CharacterString">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
```

マッピング規則

この PI タイプの符号化規則は、ISO19118 AnnexA に準拠している。

場所型のサンプル

作成した、施設型の場所型のサンプルを、表附 2-57 に示す。

実際のデータの形式は空間参照系管理サービスに依存するが、ここでは例として XML 形式での作成例を示す。

表附 2-57 施設型の場所型のサンプル (XML 形式)

以下、場所型のインスタンス部分の抜粋

```
<jci:PI_LocationType id="lt_fac">
  <jps:SI_LocationType.theme>施設の位置の指定</jps:SI_LocationType.theme>
  <jps:SI_LocationType.definition>河川及び国道事務所が管理する河川管理施設と道路管理施設、並びに公共
  と民間の施設</jps:SI_LocationType.definition>
  <jps:SI_LocationType.territoryOfUse>日本全国</jps:SI_LocationType.territoryOfUse>
  <jps:SI_LocationType.owner>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>電子地図/建設情報連携小委員会
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationType.owner>
  <jci:PI_LocationType.name>施設型</jci:PI_LocationType.name>
  <jci:PI_LocationType.identification>
    <jci:PI_Type>
      <PI_Type.applicationSchema>http://xxxxx.jp/pitype/fac.html</PI_Type.applicationSchema>
      <PI_Type.encodingRule>http://xxxxx.jp/pitype/fac.html</PI_Type.encodingRule>
      <PI_Type.mappingRule>http://xxxxx.jp/pitype/fac.html</PI_Type.mappingRule>
    </jci:PI_Type>
  </jci:PI_LocationType.identification>
</jci:PI_LocationType>
```

2-3. 地理識別子の作成

この箇条では、抽出した 12 項目の地理識別子を事例として、地理識別子の整備方法を解説する。地理識別子の作成にあたっては、既に整備されたデータがあり、それを利用すればよい場合と、整備する際に利用する既存リソースが整備する機関（例えば、整備局や事務所）が所有するものであるのか、そうでないのかによって手順は異なってくるため、「既存リソースを所有している場合の整備方法」と「既存リソースを所有していない場合の整備方法」に分けて解説する。また、作成にあたっては、附属書 2-1 で整理した整備優先度の高い地理識別子から整備するのがよい。

なお、この箇条では、「住所」と「管理路線」について、具体的な地理識別子の整備方法の例を示す。その他の地理識別子については、整備の流れと作成サンプルを簡潔に示す。

実際に地理識別子を整備する際は、「住所」と「管理路線」の整備例を参考にするとよい。

2-3-1. 既に整備されたデータを利用すればよい地理識別子の整備

道路・河川分野の 12 種類の地理識別子の中では、「住所」がこれに該当する。

(A)住所の地理識別子の整備

既存リソースの収集

住所の地理識別子の整備にあたっては、街区レベル位置参照情報が既存リソースとして考えられる。このデータは国土交通省国土計画局で整備されているもので、国土交通省国土計画局の Web サイトから無償でダウンロードをすることができる。

国土交通省国土計画局の Web サイトの URL

<http://nlftp.mlit.go.jp/isj/index.html>



図附 2-19 市区レベル位置参照情報ダウンロードサイト

ダウンロードサイトでは、都道府県単位または市町村単位でダウンロードする対象を指定し、ダウンロードすることができる。

ダウンロードした圧縮ファイルには、表附 2-58 に示すファイル群が格納されており、csv形式のファイルが、実際の住所や対応する座標値が記述された市区レベル位置参照情報のデータである。

表附 2-58 ダウンロードできる市区レベル位置参照情報のファイル群

データ	内容
「.csv」	具体的な住所やそれに対応する座標値を記述
「.html」	csv ファイル内のデータの属性項目及びその内容を記載
「.xml」	ダウンロードした市区レベル位置参照情報のメタデータ
“ ”	”に入る名称は、ダウンロードした各ファイルによって異なる

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
2	千葉県	柏市	高田	1	9	-14180.3	9780	35.87214	139.9416	0	1	0	0	0	0	0
3	千葉県	柏市	高田	4	9	-14180.3	9780	35.87214	139.9416	0	1	0	0	0	0	0
4	千葉県	柏市	高田	5	9	-14180.3	9780	35.87214	139.9416	0	1	0	0	0	0	0
5	千葉県	柏市	高田	6	9	-14180.3	9780	35.87214	139.9416	0	1	0	0	0	0	0
6	千葉県	柏市	高田	7	9	-14180.3	9780	35.87214	139.9416	0	1	0	0	0	0	0
7	千葉県	柏市	高田	8	9	-14180.3	9780	35.87214	139.9416	0	1	0	0	0	0	0
8	千葉県	柏市	高田	9	9	-14180.3	9780	35.87214	139.9416	0	1	0	0	0	0	0
9	千葉県	柏市	高田	17	9	-14030.7	9992.9	35.87349	139.944	0	1	0	0	0	0	0
10	千葉県	柏市	高田	19	9	-14030.7	9992.9	35.87349	139.944	0	1	0	0	0	0	0
11	千葉県	柏市	高田	41	9	-14030.7	9992.9	35.87349	139.944	0	1	0	0	0	0	0
12	千葉県	柏市	高田	44	9	-14030.7	9992.9	35.87349	139.944	0	1	0	0	0	0	0
13	千葉県	柏市	高田	57	9	-14004.2	9820.1	35.87373	139.9421	0	1	0	0	0	0	0
14	千葉県	柏市	高田	58	9	-14004.2	9820.1	35.87373	139.9421	0	1	0	0	0	0	0
15	千葉県	柏市	高田	63	9	-13876.3	9836.1	35.87488	139.9423	0	1	0	0	0	0	0
16	千葉県	柏市	高田	63	9	-13857.7	9897.5	35.87505	139.9429	0	0	0	0	0	0	0
17	千葉県	柏市	高田	63	9	-13827.7	9875.6	35.87532	139.9427	0	0	0	0	0	0	0
18	千葉県	柏市	高田	63	9	-13800.7	9939.6	35.87556	139.9434	0	0	0	0	0	0	0
19	千葉県	柏市	高田	66	9	-13919.2	9975.1	35.87449	139.9438	0	1	0	0	0	0	0

図附 2-20 街区レベル位置参照情報（Excel でファイルを開いた例）

地理識別子の作成方法の検討

住所の地理識別子は、ダウンロードした街区レベル位置参照のデータを用い、本編で示したデータ構造に則って場所インスタンスとして作成する。さらに、地理識別子の値（PI の値）は、場所型の中の PI タイプで定めた符号化規則に従って記述をしなければならない。また、本編 3.2 で示した決まりごとに対しても遵守して作成をしなければならない。

なお、住所のように大量の地理識別子を作成する必要がある場合は、作成したいデータ形式に応じた一括処理プログラムを作成し、データ作成を行うのがよい。

以下に、場所インスタンスの各属性・関連役割にどのように値を付与するのか、例を示す。

地理識別子（多重度：1、値の型：PI）

地理識別子の値は、“PI の値”と“PI の空間参照系”で構成される。

“PI の値”は、附属書 2-2. で示した PI タイプの符号化規則に従って記述しなければならない。なお、街区レベル位置参照情報から作成する場合の、作成できる住所の最小単位は「街区番号・地番」までであり（例：千葉県柏市あけぼの 5 丁目 1 番）住居番号や枝番を含む住所の場所インスタンス（例：千葉県柏市あけぼの 5 丁目 1 番 1 号）は作成することができない。

“PI の空間参照系”は、どの場所インスタンスに対しても、附属書 2-2-1. で示した地理識別子による空間参照系の値を付与する。

代替地理識別子（多重度：0..*、値の型：CharacterString）

代替地理識別子は記述しなければならない属性ではないが、住所の場合、以下のような地位名の“揺らぎ”が存在することから、それらの問題をデータとしても吸収するために

も、この属性に地名の“揺らぎ”のパターンを示しておくことが望ましい。

<住所で考えられる、地名の“揺らぎ”>

・詳細地名の省略

丁目や番地、番、号などの省略や、ハイフンや「ノ」等で略記される。

・アラビア数字による代用

丁目などの漢数字がアラビア数字により代用される。

・異体字の使用

常用漢字にない漢字が地名に用いられている場合に、よく似た漢字で代用される。

・小文字の省略・追加

「が」「ヶ」「ケ」「ャ」「ヤ」「ツ」「ッ」は、それぞれ混合して使用あるいは、省略、場合によっては余分に追加されることがある。

地理範囲（多重度：1、 値の型：EX_GeographicExtent）

街区レベル位置参照情報は、街区の代表点をデータとして保有しているため、地理識別子に付与する位置情報は「点」で与えるのが最も作業上平易である。したがって、この属性では“千葉県柏市”等の、この地理識別子が存在する場所をテキストで記述し、実際の座標値は、下記の「位置」でもつこととした。

位置（多重度：0..1、 値の型：GM_Point）

街区レベル位置参照情報の各データがもつ座標値を、代表点として付与した。

時間範囲（多重度：0..1、 値の型：TM_Period）

時間範囲は記述しなければならない属性ではないが、その地物の存続期間を記述しておくことで、ある時点を指定した地名の検索を行うことも可能となるため、できるだけ記述するとよい。

管理者（多重度：1、 値の型：CI_ResponsibleParty）

この属性には、この場所インスタンスを管理する整備主体を記述する。住所の場合は、各地方整備局の名称を記述する。

地名辞典（多重度：0..*、 値の型：SI_Gazetteer）

この関連役割は記述しなければならない関連役割ではないが、作成した場所インスタンスがどの地名辞典に格納されるのか、原則的に記述すべき関連役割である。

ここでは、どの場所インスタンスも各事務所で作成する地名辞典の名称を記述する。

場所型（多重度：1、 値の型：SI_LocationType）

ここでは、どの場所インスタンスも附属書 2-2-2.で示した「住所型」を必ず記述しなければならない。

以上を踏まえた、「住所」の場所インスタンスへの値の付与例を、表附 2-59 に示す。

表附 2-59 「住所」の場所インスタンスの値の付与例

属性・関連役割		値
地理識別子	値	<Jyusho> <Todofuken name="千葉県"> <City name="柏市"> <Choaza name="あけぼの五丁目"> <Gaikufugo name="1 番"/> </Choaza> </City> </Todofuken> </Jyusho>
	空間参照系	road.srs.jacic.or.jp
代替地理識別子 (この属性の記述は任意)		千葉県柏市あけぼの 5-1 ----- 千葉県柏市あけぼの 5 丁目 1 番 ----- 千葉県柏市あけぼの五丁目 1 番
地理範囲		千葉県柏市
位置		35.868775 , 139.969222
時間範囲(この属性の記述は任意)		—
管理者		関東地方整備局
地名辞典		千葉国道事務所地名辞典
場所型		住所型

地理識別子の作成

作成した住所の場所インスタンスのサンプルを、表附 2-60 に示す。

データの形式は地名辞典管理サービスによって異なってよく、例えば、Microsoft Access 等のリレーショナルデータベースにデータを格納する場合は、1 レコードの各カラムに、表附 2-59 に示したそれぞれの値が格納される。ただし、Web 空間でのデータ交換、つまり、複数のサービスを連携させた中で地名辞典を運用することを考慮すると、例えば表附 2-60 に示すように XML 形式でデータを作成することが考えられる。

表附 2-60 「住所」の場所インスタンスのサンプル (XML 形式の場合)

以下、場所インスタンスの1インスタンスの抜粋

```

<jci:PI_LocationInstance id="li_add_XXXXXX">
  <jps:SI_LocationInstance.alternativeGeographicIdentifier>千葉県柏市あけぼの5丁目1番
</jps:SI_LocationInstance.alternativeGeographicIdentifier>
  <jps:SI_LocationInstance.alternativeGeographicIdentifier>千葉県柏市あけぼの5-1
</jps:SI_LocationInstance.alternativeGeographicIdentifier>
  <jps:SI_LocationInstance.alternativeGeographicIdentifier>千葉県柏市あけぼの五丁目1番
</jps:SI_LocationInstance.alternativeGeographicIdentifier>
  <jps:SI_LocationInstance.administrator>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>関東地方整備局
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationInstance.administrator>
  <jps:SI_LocationInstance.position idref="pt_li_add_XXXXXX" />
  <jci:PI_LocationInstance.geographicIdentifier>
    <jci:PI>
      <jci:PI.value>
        <Jyusho>
          <Todofuken name="千葉県">
            <City name="柏市">
              <Choaza name="あけぼの五丁目">
                <Gaikufugo name="1番"/>
              </Choaza>
            </City>
          </Todofuken>
        </Jyusho>
      </jci:PI.value>
      <jci:PI.rs>road.srs.jacic.or.jp</jci:PI.rs >
    </jci:PI>
  </jci:PI_LocationInstance.geographicIdentifier>
  <jci:PI_LocationInstance.geographicExtent idref="exgd001" />
  <jci:PI_LocationInstance.gazetteer idref="gaz_add" />
  <jci:PI_LocationInstance.locationType idref="lt_add" />
</jci:PI_LocationInstance>
<jps:GM_Point id="pt_li_add_XXXXXX">
  <jps:GM_Point.position>
    <jps:DirectPosition>
      <DirectPosition.coordinate>35.XXXXX 139.YYYYY</DirectPosition.coordinate>
      <DirectPosition.dimension>2</DirectPosition.dimension>
      <DirectPosition.coordinateReferenceSystem idref="rs001" />
    </jps:DirectPosition>
  </jps:GM_Point.position>
</jps:GM_Point>

```

作成した地理識別子の品質のチェック

作成した地理識別子に対しては、品質評価を行わなければならない。品質評価によって、

想定した間違いの割合（これを誤率という）を超える間違いがデータの中に存在しないかどうかを確認し、想定以上の誤率があった場合には、データの修正を行わなければならない。品質評価においても、地理識別子の作成の場合と同じく品質評価ツールを作成し、一括処理によって評価を行うのが最も効率がよいと考えられる。

さらに、品質評価の結果は、地名辞典に記述しておくことが望ましい。

を踏まえ、地理識別子を格納する地名辞典のサンプルを表附 2-61 に示す。

表附 2-61 「住所」の地名辞典のサンプル

以下、地名辞典のインスタンス部分の抜粋

```
<jci:EXPI_Gazetteer id="gaz_add">
  <jps:SI_Gazetteer.scope>この地名辞典は、  地方整備局内の住所を整備したものである
</jps:SI_Gazetteer.scope >
  <jps:SI_Gazetteer.territoryOfUse idref="exgd_gazadd_XXXXXX " />
  <jps:SI_Gazetteer.custodian>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName>  地方整備局
    </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_Gazetteer.custodian >
  <jci:PI_Gazetteer.name>http://www.  .go.jp/gaz/gaz_add</jci:PI_Gazetteer.name >
  <jci:PI_Gazetteer.date idref="inst_XXXXXX " />
  <jci:PI_Gazetteer.alias>  地方整備局住所地名辞典</jci:PI_Gazetteer.alias >
  <jci:PI_Gazetteer.quality idref="dq_XXXXXX " />
  <jci:EXPI_Gazetteer.lrs idref="lrs_road " />
  <jci:EXPI_Gazetteer.locationInstances idref=" li_add_XXXXXX " />
  <jci:EXPI_Gazetteer.locationInstances idref=" li_add_YYYYYY " />
  <jci:EXPI_Gazetteer.locationInstances idref=" li_add_ZZZZZZ " />
```

以下、この地名辞典が格納する場所インスタンスをすべて列挙する

```
</jci:EXPI_Gazetteer>
<jps:EX_GeographicDescription id="exgd_gazadd_XXXXXX">
  <jps:EX_GeographicDescription.geographicIdentifier>  地方整備局内
</jps:EX_GeographicDescription.geographicIdentifier>
</jps:EX_GeographicDescription>
<jps:TM_Instant id="inst_XXXXXX ">
  <jps:TM_Instant.position>
    <jps:TM_CalDate>
      <TM_TemporalPosition.frame idref="rs002"/>
      <TM_CalDate.calDate>2007 3 29</TM_CalDate.calDate>
      <TM_CalDate.calendarEraName>西暦</TM_CalDate.calendarEraName>
    </jps:TM_CalDate>
  </jps:TM_Instant.position>
</jps:TM_Instant>
```

住所の地名辞典作成に掛かる作業量

～ で各作業工程で掛かる作業量は、概ね以下の通りである。

既存リソースの収集：1日

地理識別子の作成方法の検討：2～3日

（変換ツールの作成を含む。ただし、作業量はツール作成者のスキルに依るところが大きい）

地理識別子の作成：0.5日

地理識別子の品質のチェック：3～5日

（変換ツールの作成を含む。ただし、作業量はツール作成者のスキルや、品質評価結果に依るところが大きい。特に、想定を超える誤率があった場合は、データの修正が必要となるため、上記の作業量を超える可能性もある）

「住所」の地理識別子の整備方法のまとめ

上記で紹介した整備方法例の流れをまとめると、表附 2-62 のようになる。

なお、上記では紹介していないが、整備した地名辞典は、地名辞典管理サービスに登録して、初めて利用が可能になる（表中の ）。この部分に関しては、附属書 2-4 で別途解説する。

表附 2-62 「住所」の地名辞典の整備方法の流れ

概要	整備する地理識別子	住所			
	原典資料	街区レベル位置参照情報、市販デジタル住宅地図 等			
	整備者	地方整備局			
	整備優先度	必要性	高い	しやすさ	整備しやすい
	位置情報の形状	点(属性 position で記述)			
手順	<p>【初期整備(街区レベル位置参照情報を既存リソースとする場合)】</p> <p>「街区レベル位置参照情報ダウンロードサービス」から地方整備局の範囲を選択し、ダウンロードする。</p> <p>街区レベル位置参照情報データ(CSV 形式)から地理識別子を作成する方法を検討し、変換ツールを作成する。</p> <p>で作成したツールを作成して、地理識別子を作成する。</p> <p>作成した地理識別子の品質評価を行う。</p> <p>作成した地理識別子を、地名辞典として地名辞典管理サービスに登録する。</p>				

2-3-2. 自分の機関で所有する既存リソースを用いた地理識別子の整備

対象となる地理識別子は「住所」以外の全ての地理識別子である。これらの地理識別子は、自分の機関で所有している既存リソースを用いて、地理識別子を作成する。

なお、作成した地理識別子には個人情報が含まれていることがあるかもしれない。整備した地名辞典は一般にも公開することを前提としており、また、利用者権限等を設定することで、地名辞典を利用できる対象を制限する仕組みについてはまだ検討がなされていないため、現時点では、作成した地名辞典の中に個人情報が含まれないように配慮する必要がある。

(A) 「管理路線」の地理識別子の整備

既存リソースの収集

「管理路線」の地理識別子の既存リソースとしては、「DRM」「数値地図」「国土数値情報」「管内図」等が考えられる。ただし、管内図は紙媒体であることがほとんどのため、整備にあたっては、スキャニングやデジタイズ等の作業が発生するため、デジタルデータ化されている「DRM」「数値地図」「国土数値情報」をリソースとして行った方がよい。

「DRM」を既存リソースとする場合、DRM を利用する場合、DRM のデータをテキスト形式や、GIS で読み込める形式に変換するツールを作成することで、地理識別子作成に必要な情報を抽出できるようになる。

地理識別子の作成方法の検討

管理路線の地理識別子は、DRM データから抽出した管理路線に関する情報を用い、本編で示したデータ構造に則って場所インスタンスとして作成する。さらに、地理識別子の値（PI の値）は、場所型の中の PI タイプで定めた符号化規則に従って記述をしなければならない。また、本編 3.2 で示した決まりごとに対しても遵守して作成をしなければならない。

以下に、場所インスタンスの各属性・関連役割にどのように値を付与するのか、例を示す。

地理識別子（多重度：1、 値の型：PI）

地理識別子の値は、“PI の値”と“PI の空間参照系”で構成される。

“PI の値”は、附属書 2-2. で示した PI タイプの符号化規則に従って記述しなければならない。

“PI の空間参照系”は、どの場所インスタンスに対しても、附属書 2-2-1. で示した地理識別子による空間参照系の値を付与する。

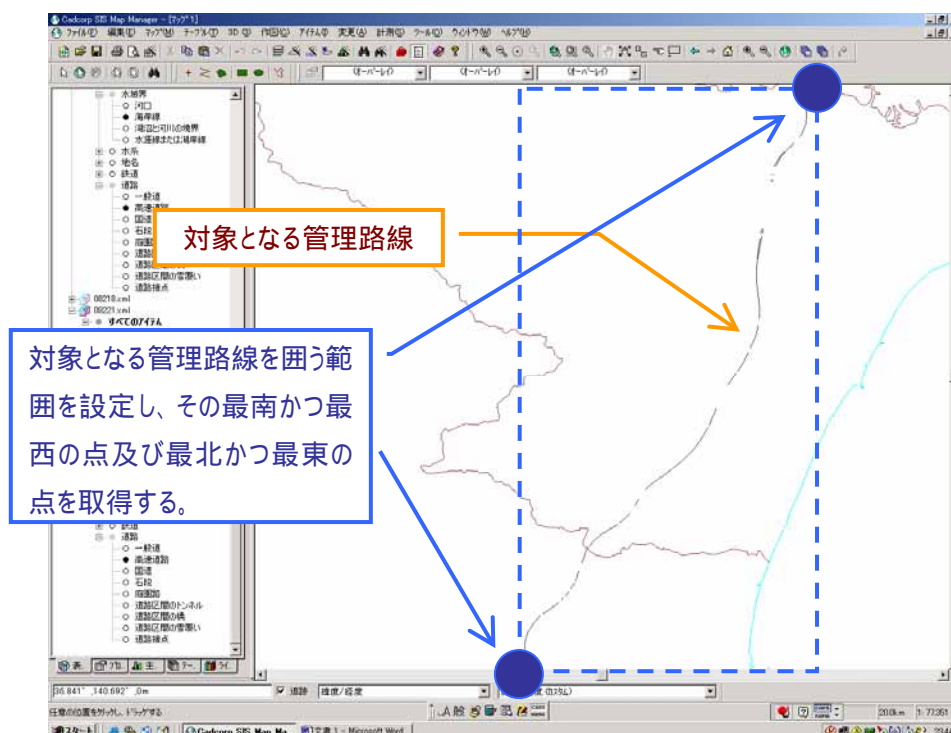
代替地理識別子（多重度：0..*、 値の型：CharacterString）

代替地理識別子は記述しなければならない属性ではないが、数字の半角と全角（“国道 16 号”と“国道 1 6 号”）等の言葉の“揺らぎ”をこの属性を利用することで吸収できるため、非常に有効である。また、管理路線には“国道 20 号”等の正式名称とともに“甲州街道”等の通称が別に使われていることも少なくない。このような場合に、代替地理識別子として“甲州街道”等の通称を記述しておくこと、データを重複して作成することなく正式名称と通称を同じ路線として扱うことができる。

地理範囲（多重度：1、 値の型：EX_GeographicExtent）

DRM のデータを GIS で読み込める形式に変換した場合、GIS の画面上で対象となる路線全体を表示させ、路線を囲う範囲の最南かつ最西の点及び最北及び最東の点を取得するの

がよい。



図附 2-21 管理路線の地理範囲の取得方法

位置（多重度：0..1、 値の型：GM_Point）

地理識別子の位置情報を地理範囲で取得したので、この属性は用いない。

時間範囲（多重度：0..1、 値の型：TM_Period）

時間範囲は記述しなければならない属性ではないが、その地物の存続期間を記述しておくことで、ある時点を指定した地名の検索を行うことも可能となるため、できるだけ記述するとよい。

管理者（多重度：1、 値の型：CI_ResponsibleParty）

この属性には、この場所インスタンスを管理する整備主体である各国道事務所の担当部署名を記述する。

地名辞典（多重度：0..*、 値の型：SI_Gazetteer）

この関連役割は記述しなければならない関連役割ではないが、作成した場所インスタンスがどの地名辞典に格納されるのか、原則的に記述すべき関連役割である。

ここでは、どの場所インスタンスも各事務所で作成する地名辞典の名称を記述する。

場所型（多重度：1、 値の型：SI_LocationType）

ここでは、どの場所インスタンスも附属書 2-2-2.で示した「管理路線型」を必ず記述しなければならない。

表附 2-63 場所インスタンスの値の付与例（管理路線）

属性・関連役割		値
地理識別子	値	<Rosen name="国道 20 号線">
	空間参照系	road.srs.jacic.or.jp
代替地理識別子 (この属性の記述は任意)		国道20号線

		国道 20 号

		国道20号

		甲州街道
地理範囲	最西の経度	140.285797

	最東の経度	140.710144

	最南の緯度	36.165597

	最北の緯度	36.763902
位置		—
時間範囲(この属性の記述は任意)		—
管理者		関東地方整備局
地名辞典		千葉国道事務所地名辞典
場所型		管理路線型

地理識別子の作成

作成した地理識別子のサンプルを表附 2-64 に示す。

データの形式は、Web 空間でのデータ交換、つまり、複数のサービスを連携させた中で地名辞典を運用することを考慮すると、例えば、表附 2-64 に示すように XML 形式でデータを作成することが考えられる。

表附 2-64 「管理路線」の地理識別子のサンプル (XML 形式で表した場合)

以下、場所インスタンスの 1 インスタンスの抜粋

```

<jci:PI_LocationInstance id="li_rd_XXXXXX">
  <jps:SI_LocationInstance.alternativeGeographicIdentifier>国道20号線
</jps:SI_LocationInstance.alternativeGeographicIdentifier>
  <jps:SI_LocationInstance.alternativeGeographicIdentifier>国道20号
</jps:SI_LocationInstance.alternativeGeographicIdentifier>
  <jps:SI_LocationInstance.alternativeGeographicIdentifier>国道20号
</jps:SI_LocationInstance.alternativeGeographicIdentifier>
  <jps:SI_LocationInstance.alternativeGeographicIdentifier>甲州街道
</jps:SI_LocationInstance.alternativeGeographicIdentifier>
  <jps:SI_LocationInstance.administrator>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName> 国道事務所
    </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001</CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationInstance.administrator>
<jci:PI_LocationInstance.geographicIdentifier>
  <jci:PI>
    <jci:PI.value>
      <Rd name="国道20号線">
    </jci:PI.value>
    <jci:PI.rs>road.srs.jacic.or.jp</jps:PI.rs >
  </jci:PI>
</jci:PI_LocationInstance.geographicIdentifier>
<jci:PI_LocationInstance.geographicExtent idref="exgb_rd_XXXXXX " />
<jci:PI_LocationInstance.gazetteer idref="gaz_rd" />
<jci:PI_LocationInstance.locationType idref="lt_rd" />
</jci:PI_LocationInstance>
<jps:EX_GeographicBoundingBox id="exgb_rd_XXXXXX">
  <jps:EX_GeographicBoundingBox.westBoundLongitude>140.XXXXX
</jps:EX_GeographicBoundingBox.westBoundLongitude>
  <jps:EX_GeographicBoundingBox.eastBoundLongitude>140.YYYYY
</jps:EX_GeographicBoundingBox.eastBoundLongitude>
  <jps:EX_GeographicBoundingBox.southBoundLongitude>36.XXXXX
</jps:EX_GeographicBoundingBox.southBoundLongitude>
  <jps:EX_GeographicBoundingBox.northBoundLongitude>36.YYYYY
</jps:EX_GeographicBoundingBox.northBoundLongitude>
</jps:EX_GeographicBoundingBox>

```

作成した地理識別子の品質のチェック

作成した地理識別子に対しては、品質評価を行わなければならない。品質評価によって、想定した間違いの割合（これを誤率という）を超える間違いがデータの中に存在しないかどうかを確認し、想定以上の誤率があった場合には、データの修正を行わなければならない。品質評価においても、地理識別子の作成の場合と同じく品質評価ツールを作成し、一括処理によって評価を行うのが最も効率がよいと考えられる。

さらに、品質評価の結果は、地名辞典に記述しておくことが望ましい。

を踏まえ、地理識別子を格納する地名辞典のサンプルを表附 2-65 に示す。

表附 2-65 「管理路線」の地名辞典のサンプル

以下、地名辞典のインスタンス部分の抜粋

```
<jci:EXPI_Gazetteer id="gaz_rd">
  <jps:SI_Gazetteer.scope>この地名辞典は、 国道事務所内の管理路線を整備したものである
</jps:SI_Gazetteer.scope >
  <jps:SI_Gazetteer.territoryOfUse idref="exgd_gazrd_XXXXXX " />
  <jps:SI_Gazetteer.custodian>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName> 国道事務所
    </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001 </CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_Gazetteer.custodian >
  <jci:PI_Gazetteer.name>http://www. .go.jp/gaz/gaz_rd</jci:PI_Gazetteer.name >
  <jci:PI_Gazetteer.date idref="inst_XXXXXX " />
  <jci:PI_Gazetteer.alias> 国道事務所管理路線地名辞典</jci:PI_Gazetteer.alias >
  <jci:EXPI_Gazetteer.quality idref="dq_XXXXXX " />
  <jci:EXPI_Gazetteer.lrs idref="lrs_road " />
  <jci:EXPI_Gazetteer.locationInstances idref=" li_rd_XXXXXX " />
  <jci:EXPI_Gazetteer.locationInstances idref=" li_rd_YYYYYY " />
  <jci:EXPI_Gazetteer.locationInstances idref=" li_rd_ZZZZZZ " />
```

以下、この地名辞典が格納する場所インスタンスをすべて列挙する

```
</jci:EXPI_Gazetteer>
<jps:EX_GeographicDescription id="exgd_gazrd_XXXXXX">
  <jps:EX_GeographicDescription.geographicIdentifier> 国道事務所内
</jps:EX_GeographicDescription.geographicIdentifier>
</jps:EX_GeographicDescription>
<jps:TM_Instant id="inst_XXXXXX ">
  <jps:TM_Instant.position>
    <jps:TM_CalDate>
      <TM_TemporalPosition.frame idref="rs002"/>
      <TM_CalDate.calDate>2007 3 29</TM_CalDate.calDate>
      <TM_CalDate.calendarEraName>西暦</TM_CalDate.calendarEraName>
    </jps:TM_CalDate>
  </jps:TM_Instant.position>
</jps:TM_Instant>
```

「管理路線」の地名辞典作成に掛かる作業量

～ で各作業工程で掛かる作業量は、概ね以下の通りである。

既存リソースの収集：1日

地理識別子の作成方法の検討：0.5～2日

（ただし、作業量は作成する管理路線の地理識別子の個数によって変動する）

地理識別子の作成：0.5～2日

（ただし、作業量は作成する管理路線の地理識別子の個数によって変動する）

地理識別子の品質のチェック：0.5～2日

（ただし、作業量は作成する管理路線の地理識別子の個数によって変動する）

「管理路線」の地理識別子の整備方法のまとめ

上記で紹介した整備方法例の流れとしてまとめると、表附 2-66 のようになる。

なお、上記では紹介していないが、整備した地名辞典は、地名辞典管理サービスに登録して、初めて利用が可能になる（表中の ）。この部分に関しては、附属書 2-4 で別途解説する。

表附 2-66 「管理路線」の地名辞典の整備方法の流れ

概要	整備する地理識別子	管理路線			
	原典資料	DRM、数値地図、国土数値情報、管内図			
	整備者	各国道事務所			
	整備優先度	必要性	高い	しやすさ	注意必要
	位置情報の形状	面(属性 geographicExtent で記述)			
手順	<p>【初期整備(DRMを既存リソースとする場合)】</p> <p>DRMのデータを、変換ツールによって、テキスト形式またはGISで読み込める形式に変換し、管理路線を示すデータ及び対応する位置情報のデータを抽出する。</p> <p>で抽出したデータを用いて、地理識別子を作成する。</p> <p>作成した地理識別子の品質評価を行う。</p> <p>作成した地理識別子を、地名辞典として地名辞典管理サービスに登録する。</p>				
<pre> graph TD A[DRMデータ] -- "変換、抽出" --> B["抽出したデータ (管理路線)"] B --> C[地理識別子] C -- "整備者" --> D[品質評価] D --> E["地名辞典"] E -- "登録" --> F["地名辞典管理サービス"] </pre>					

以降は、地理識別子ごとに整備の流れと作成サンプルを簡潔に示す。

場所インスタンスに付与する値の考え方や、場所インスタンスを格納する地名辞典の作成方法については、前述の「住所」「管理路線」を参考にするとよい。

(B) 道路距離標の地理識別子の整備

「道路距離標」の地理識別子の既存リソースとしては、「DRM」や「道路管内図」が考えられる。ただし、道路管内図は紙媒体であることがほとんどのため、DRM を利用するのがよい。DRM を利用する場合、DRM のデータをテキスト形式や、GIS で読み込める形式に変換するツールを作成することで、地理識別子作成に必要な情報を抽出できるようになる。

また、地理識別子に付与する位置情報の形状は、「点」とするのがよい。

表附 2-67 に、「道路距離標」の地理識別子の整備方法の概要と流れを示す。

表附 2-67 「道路距離標」の地名辞典の整備方法の流れ

概要	整備する地理識別子	道路距離標			
	原典資料	DRM、道路管内図			
	整備者	各国道事務所			
	整備優先度	必要性	高い	しやすさ	整備しやすい
	位置情報の形状	点(属性 position で記述)			
手順	<p>【初期整備(DRM を既存リソースとする場合)】</p> <p>DRM のデータを、変換ツールによって、テキスト形式または GIS で読み込める形式に変換し、交差点を示すデータ及び対応する位置情報のデータを抽出する。</p> <p>で抽出したデータを用いて、地理識別子を作成する。</p> <p>作成した地理識別子の品質評価を行う。</p> <p>作成した地理識別子を、地名辞典として地名辞典管理サービスに登録する。</p> <pre> graph TD DRM[DRM データ] -- "変換、抽出" --> Data[抽出したデータ (道路距離標)] Data --> Prep[整備者] Prep -- "品質評価" --> Dict[地名辞典] Prep -- "登録" --> Service[地名辞典管理サービス] </pre>				

また、「道路距離標」の地理識別子の作成サンプル (XML 形式で作成した場合) を、表附 2-68 に示す。

表附 2-68 「道路距離標」の地理識別子の作成サンプル

以下、場所インスタンスの1インスタンスの抜粋

```
<jci:PI_LocationInstance id="li_rdkp_XXXXXX">
  <jps:SI_LocationInstance.administrator>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName> 国道事務所
    </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001 </CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationInstance.administrator>
  <jps:SI_LocationInstance.position idref="pt_li_rdkp_XXXXXX" />
  <jci:PI_LocationInstance.geographicIdentifier>
    <jci:PI>
      <PI.value>
        <RdKp>
          <Rd type="国道" name=" 号線">
            <Kp type="上り" name="11.0kp"/>
          </Rd>
        </RdKp>
      </PI.value>
      <PI.rs>road.srs.jacic.or.jp</PI.rs >
    </jci:PI>
  </jci:PI_LocationInstance.geographicIdentifier>
  <jci:PI_LocationInstance.gazetteer idref="gaz_rdkp" />
  <jci:PI_LocationInstance.locationType idref="lt_rdkp" />
</jci:PI_LocationInstance>
<jps:GM_Point id="pt_li_rdkp_XXXXXX">
  <jps:GM_Point.position>
    <jps:DirectPosition>
      <DirectPosition.coordinate>35.XXXXX 139. XXXXX</DirectPosition.coordinate>
      <DirectPosition.dimension>2</DirectPosition.dimension>
      <DirectPosition.coordinateReferenceSystem idref="rs001" />
    </jps:DirectPosition>
  </jps:GM_Point.position>
</jps:GM_Point>
```

(C) 交差点の地理識別子の整備

「交差点」の地理識別子の既存リソースとしては、「DRM」が考えられる。DRM のデータをテキスト形式や、GIS で読み込める形式に変換するツールを作成することで、地理識別子作成に必要な情報を抽出できるようになる。

また、地理識別子に付与する位置情報の形状は、「点」とするのがよい。

表附 2-69 に、「交差点」の地理識別子の整備方法の概要と流れを示す。

表附 2-69 「交差点」の地理識別子の整備の概要

概要	整備する地理識別子	交差点			
	原典資料	DRM			
	整備者	国道事務所			
	整備優先度	必要性	高くない	しやすさ	整備しやすい
	位置情報の形状	点(属性 position で記述)			
手順 (例)	<p>【初期整備(DRM を既存リソースとする場合)】</p> <p>DRM のデータを、変換ツールによって、テキスト形式または GIS で読み込める形式に変換し、交差点を示すデータ及び対応する位置情報のデータを抽出する。</p> <p>で抽出したデータを用いて、地理識別子を作成する。</p> <p>作成した地理識別子の品質評価を行う。</p> <p>作成した地理識別子を、地名辞典として地名辞典管理サービスに登録する。</p> <pre> graph TD DRM[DRM データ] -- "変換、抽出" --> Data[抽出したデータ (交差点)] Data --> Preparer[整備者] Preparer -- "品質評価" --> Dict[地名辞典] Dict -- "登録" --> Service[地名辞典管理サービス] </pre>				

また、「交差点」の地理識別子の作成サンプル（XML 形式で作成した場合）を、表附 2-70 に示す。

表附 2-70 「交差点」の地理識別子の作成サンプル

以下、場所インスタンスの 1 インスタンスの抜粋

```
<jci:PI_LocationInstance id="li_its_XXXXXX">
  <jps:SI_LocationInstance.alternativeGeographicIdentifier> 交差点
  </jps:SI_LocationInstance.alternativeGeographicIdentifier>
  <jps:SI_LocationInstance.administrator>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName> 国道事務所
    </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001 </CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationInstance.administrator>
  <jps:SI_LocationInstance.position idref="pt_li_its_XXXXXX" />
  <jci:PI_LocationInstance.geographicIdentifier>
    <jci:PI>
      <PI.value>
        <Its>
          <Rd type="国道" name=" 号線">
            <ItsN name=" 交差点" />
          </Rd>
        </Its>
      </PI.value>
      <PI.rs>road.srs.jacic.or.jp</PI.rs >
    </jci:PI>
  </jci:PI_LocationInstance.geographicIdentifier>
  <jci:PI_LocationInstance.gazetteer idref="gaz_its" />
  <jci:PI_LocationInstance.locationType idref="lt_its" />
</jci:PI_LocationInstance>
<jps:GM_Point id="pt_li_its_XXXXXX">
  <jps:GM_Point.position>
    <jps:DirectPosition>
      <DirectPosition.coordinate>35.XXXXX 139. XXXXX</DirectPosition.coordinate>
      <DirectPosition.dimension>2</DirectPosition.dimension>
      <DirectPosition.coordinateReferenceSystem idref="rs001" />
    </jps:DirectPosition>
  </jps:GM_Point.position>
</jps:GM_Point>
```

(D) 管理河川の地理識別子の整備

「管理河川」の地理識別子の既存リソースとしては、「河川基盤地図」「数値地図」「国土数値情報」「管内図」等が考えられる。ただし、管内図は紙媒体であることがほとんどのため、整備にあたっては、スキャンやデジタル化等の作業が発生するため、デジタルデータ化されている「河川基盤地図」「数値地図」「国土数値情報」をリソースとして行った方がよい。

また、地理識別子に付与する位置情報の形状は、「面」とするのがよい。表附 2-71 に、「管理河川」の地理識別子の整備方法の概要と流れを示す。

表附 2-71 「管理河川」の地名辞典の整備方法の流れ

概要	整備する地理識別子	管理河川			
	原典資料	河川基盤地図、数値地図、国土数値情報、管内図			
	整備者	各河川事務所			
	整備優先度	必要性	高い	しやすさ	注意必要
	位置情報の形状	面(属性 geographicExtent で記述)			
手順	<p>【初期整備(河川基盤地図を既存リソースとする場合)】</p> <p>河川基盤地図から、河川距離標を示すデータ及び対応する位置情報のデータを抽出する。</p> <p>で抽出したデータを用いて、地理識別子を作成する。</p> <p>作成した地理識別子の品質評価を行う。</p> <p>作成した地理識別子を、地名辞典として地名辞典管理サービスに登録する。</p>				

また、「管理河川」の地理識別子の作成サンプル（XML 形式で作成した場合）を、表附 2-72 に示す。

表附 2-72 「管理河川」の地理識別子の作成サンプル

以下、場所インスタンスの 1 インスタンスの抜粋

```
<jci:PI_LocationInstance id="li_riv_XXXXXX">
  <jps:SI_LocationInstance.alternativeGeographicIdentifier>利根川
</jps:SI_LocationInstance.alternativeGeographicIdentifier>
  <jps:SI_LocationInstance.alternativeGeographicIdentifier>とねがわ
</jps:SI_LocationInstance.alternativeGeographicIdentifier>
  <jps:SI_LocationInstance.administrator>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName> 河川事務所
    </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001 </CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationInstance.administrator>
  <jci:PI_LocationInstance.geographicIdentifier>
    <jci:PI>
      <PI.value>
        <Riv type="一級河川" name="利根川"/>
      </PI.value>
      <PI.rs>river.srs.jacic.or.jp</PI.rs >
    </jci:PI>
  </jci:PI_LocationInstance.geographicIdentifier>
  <jci:PI_LocationInstance.geographicExtent idref="exgb_riv_XXXXXX" />
  <jci:PI_LocationInstance.gazetteer idref="gaz_riv" />
  <jci:PI_LocationInstance.locationType idref="lt_riv" />
</jci:PI_LocationInstance>
<jps:EX_GeographicBoundingBox id="exgb_riv_XXXXXX">
  <jps:EX_GeographicBoundingBox.westBoundLongitude>140.XXXXX
</jps:EX_GeographicBoundingBox.westBoundLongitude>
  <jps:EX_GeographicBoundingBox.eastBoundLongitude>140.YYYYY
</jps:EX_GeographicBoundingBox.eastBoundLongitude>
  <jps:EX_GeographicBoundingBox.southBoundLongitude>36.XXXXX
</jps:EX_GeographicBoundingBox.southBoundLongitude>
  <jps:EX_GeographicBoundingBox.northBoundLongitude>36.YYYYY
</jps:EX_GeographicBoundingBox.northBoundLongitude>
</jps:EX_GeographicBoundingBox>
```

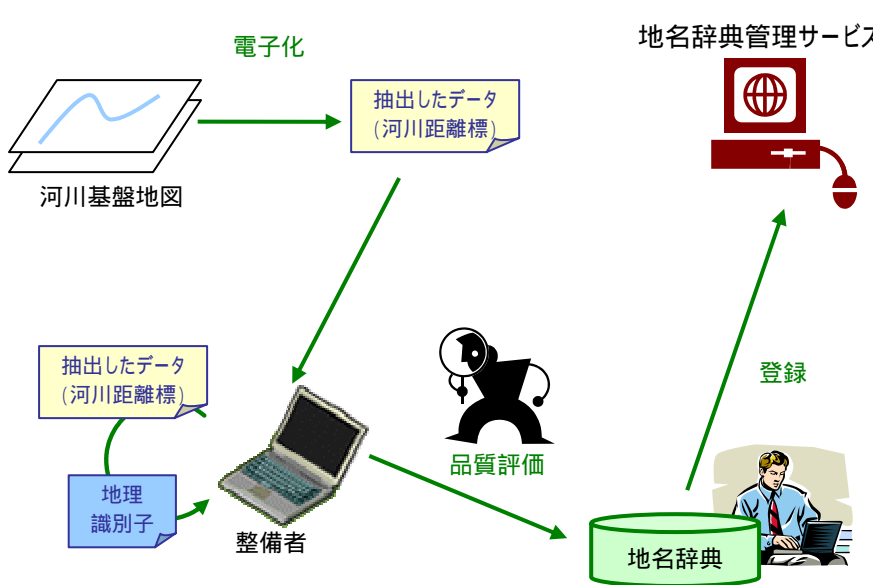
(E) 「河川距離標」の地理識別子の整備

「河川距離標」の地理識別子の既存リソースとしては、「河川基盤地図」や「管内図」が考えられる。ただし、管内図は紙媒体であることがほとんどのため、整備にあたっては、スキャニングやデジタイズ等の作業が発生するため、「河川基盤地図」があれば、これをリソースとして行った方がよい。

また、地理識別子に付与する位置情報の形状は、「点」とするのがよい。

表附 2-73 に、「河川距離標」の地理識別子の整備方法の概要と流れを示す。

表附 2-73 「河川距離標」の地理識別子の整備の概要

概要	整備する地理識別子	河川距離標			
	原典資料	河川基盤地図、管内図			
	整備者	河川事務所			
	整備優先度	必要性	高くない	しやすさ	整備しやすい
	位置情報の形状	点(属性 position で記述)			
手順 (例)	<p>【初期整備(河川基盤地図を既存リソースとする場合)】</p> <p>河川基盤地図から、河川距離標を示すデータ及び対応する位置情報のデータを抽出する。</p> <p>で抽出したデータを用いて、地理識別子を作成する。</p> <p>作成した地理識別子の品質評価を行う。</p> <p>作成した地理識別子を、地名辞典として地名辞典管理サービスに登録する。</p>  <p>The flowchart illustrates the process: 1. '河川基盤地図' (River Base Map) is digitized ('電子化') into '抽出したデータ (河川距離標)' (Extracted Data). 2. This data is used by the '整備者' (Preparer) on a laptop to create '地理識別子' (Geographic Identifiers). 3. The identifiers undergo '品質評価' (Quality Evaluation). 4. The final data is registered ('登録') into the '地名辞典管理サービス' (Place Name Dictionary Management Service), which is linked to a '地名辞典' (Place Name Dictionary) database.</p>				

また、「河川距離標」の地理識別子の作成サンプル (XML 形式で作成した場合) を、表附 2-74 に示す。

表附 2-74 「河川距離標」の地理識別子の作成サンプル

以下、場所インスタンスの 1 インスタンスの抜粋

```
<jci:PI_LocationInstance id="li_rivkp_XXXXXX">
  <jps:SI_LocationInstance.administrator>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName> 河川事務所
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001 </CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationInstance.administrator>
  <jps:SI_LocationInstance.position idref="pt_li_rivkp_XXXXXX" />
  <jci:PI_LocationInstance.geographicIdentifier>
    <jci:PI>
      <PI.value>
        <RivKp>
          <Riv type="一級河川" name=" 川">
            <Kp type="右岸" name="11.0kp"/>
          </Riv>
        </RivKp>
      </PI.value>
      <PI.rs>river.srs.jacic.or.jp</PI.rs >
    </jci:PI>
  </jci:PI_LocationInstance.geographicIdentifier>
  <jci:PI_LocationInstance.gazetteer idref="gaz_rivkp" />
  <jci:PI_LocationInstance.locationType idref="lt_rivkp" />
</jci:PI_LocationInstance>
<jps:GM_Point id="pt_li_rivkp_XXXXXX">
  <jps:GM_Point.position>
    <jps:DirectPosition>
      <DirectPosition.coordinate>35.XXXXX 139. XXXXX</DirectPosition.coordinate>
      <DirectPosition.dimension>2</DirectPosition.dimension>
      <DirectPosition.coordinateReferenceSystem idref="rs001" />
    </jps:DirectPosition>
  </jps:GM_Point.position>
</jps:GM_Point>
```

(F) 「水系」の地理識別子の整備

「水系」の地理識別子の既存リソースとしては、「管内図」が考えられる。ただし、管内図は紙媒体であることがほとんどのため、整備にあたっては、スキャニングやデジタル化等の作業が発生する。

また、地理識別子に付与する位置情報の形状は、「面」とするのがよい。

表附 2-75 に、「水系」の地理識別子の整備方法の概要と流れを示す。

表附 2-75 「水系」の地理識別子の整備の概要

概要	整備する地理識別子	水系			
	原典資料	管内図			
	整備者	河川事務所			
	整備優先度	必要性	低い	しやすさ	整備しづらい
	位置情報の形状	面(属性 geographicExtent で記述)			
手順 (例)	<p>【初期整備(管内図(紙媒体)より作成)】</p> <p>管内図をスキャニング等によって、電子データにする。</p> <p>GIS 上で、デジタル化等によって管轄区間をデータ入力する。</p> <p>で作成したデータから、地理識別子を作成する。</p> <p>作成した地理識別子の品質評価を行う。</p> <p>作成した地理識別子を、地名辞典として地名辞典管理サービスに登録する。</p>				

また、「水系」の地理識別子の作成サンプル（XML 形式で作成した場合）を、表附 2-76 に示す。

表附 2-76 「水系」の地理識別子の作成サンプル

```
以下、場所インスタンスの1インスタンスの抜粋
<jci:PI_LocationInstance id="li_rivra_XXXXXX">
  <jps:SI_LocationInstance.administrator>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName> 河川事務所
    </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001 </CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationInstance.administrator>
  <jci:PI_LocationInstance.geographicIdentifier>
    <jci:PI>
      <PI.value>
        <RivRa type="一級水系" name=" 川水系"/>
      </PI.value>
      <PI.rs>river.srs.jacic.or.jp</PI.rs >
    </jci:PI>
  </jci:PI_LocationInstance.geographicIdentifier>
  <jci:PI_LocationInstance.geographicExtent idref="exgb_rivra_XXXXXX" />
  <jci:PI_LocationInstance.gazetteer idref="gaz_rivra" />
  <jci:PI_LocationInstance.locationType idref="lt_rivra" />
</jci:PI_LocationInstance>
<jps:EX_GeographicBoundingBox id="exgb_rivra_XXXXXX">
  <jps:EX_GeographicBoundingBox.westBoundLongitude>140.XXXXXX
</jps:EX_GeographicBoundingBox.westBoundLongitude>
  <jps:EX_GeographicBoundingBox.eastBoundLongitude>140.YYYYYY
</jps:EX_GeographicBoundingBox.eastBoundLongitude>
  <jps:EX_GeographicBoundingBox.southBoundLongitude>36.XXXXXX
</jps:EX_GeographicBoundingBox.southBoundLongitude>
  <jps:EX_GeographicBoundingBox.northBoundLongitude>36.YYYYYY
</jps:EX_GeographicBoundingBox.northBoundLongitude>
</jps:EX_GeographicBoundingBox>
```

(G)「流域」の地理識別子の整備

「流域」の地理識別子の既存リソースとしては、「管内図」が考えられる。ただし、管内図は紙媒体であることがほとんどのため、整備にあたっては、スキャニングやデジタル化等の作業が発生する。

また、地理識別子に付与する位置情報の形状は、「面」とするのがよい。

表附 2-77 に、「流域」の地理識別子の整備方法の概要と流れを示す。

表附 2-77 「流域」の地理識別子の整備の概要

概要	整備する地理識別子	流域			
	原典資料	管内図			
	整備者	河川事務所			
	整備優先度	必要性	高くない	しやすさ	整備しづらい
	位置情報の形状	面(属性 geographicExtent で記述)			
手順 (例)	<p>【初期整備(管内図(紙媒体)より作成)】</p> <p>管内図をスキャニング等によって、電子データにする。</p> <p>GIS 上で、デジタル化等によって管轄区間をデータ入力する。</p> <p>で作成したデータから、地理識別子を作成する。</p> <p>作成した地理識別子の品質評価を行う。</p> <p>作成した地理識別子を、地名辞典として地名辞典管理サービスに登録する。</p>				

また、「流域」の地理識別子の作成サンプル（XML 形式で作成した場合）を、表附 2-78 に示す。

表附 2-78 「流域」の地理識別子の作成サンプル

```
以下、場所インスタンスの1インスタンスの抜粋
<jci:PI_LocationInstance id="li_rivws_XXXXXX">
  <jps:SI_LocationInstance.administrator>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName> 河川事務所
      </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001 </CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationInstance.administrator>
  <jci:PI_LocationInstance.geographicIdentifier>
    <jci:PI>
      <PI.value>
        <RivWs name=" 川流域"/>
      </PI.value>
      <PI.rs>river.srs.jacic.or.jp</PI.rs >
    </jci:PI>
  </jci:PI_LocationInstance.geographicIdentifier>
  <jci:PI_LocationInstance.geographicExtent idref="exgb_rivws_XXXXXX" />
  <jci:PI_LocationInstance.gazetteer idref="gaz_rivws" />
  <jci:PI_LocationInstance.locationType idref="lt_rivws" />
</jci:PI_LocationInstance>
<jps:EX_GeographicBoundingBox id="exgb_rivws_XXXXXX">
  <jps:EX_GeographicBoundingBox.westBoundLongitude>140.XXXXX
</jps:EX_GeographicBoundingBox.westBoundLongitude>
  <jps:EX_GeographicBoundingBox.eastBoundLongitude>140.YYYYY
</jps:EX_GeographicBoundingBox.eastBoundLongitude>
  <jps:EX_GeographicBoundingBox.southBoundLongitude>36.XXXXX
</jps:EX_GeographicBoundingBox.southBoundLongitude>
  <jps:EX_GeographicBoundingBox.northBoundLongitude>36.YYYYY
</jps:EX_GeographicBoundingBox.northBoundLongitude>
</jps:EX_GeographicBoundingBox>
```

(H) 「測点」の地理識別子の整備

「測点」の地理識別子の既存リソースとしては、河川分野・道路分野ともに「工事図面」が考えられる。ただし、工事図面は紙媒体である場合も多く、整備にあたっては、スキャニングやデジタイズ等の作業が発生する。

また、地理識別子に付与する位置情報の形状は、「点」とするのがよい。

表附 2-79 に、「測点」の地理識別子の整備方法の概要と流れを示す。

表附 2-79 「測点」の地理識別子の整備の概要

概要	整備する地理識別子	測点			
	原典資料	工事図面(河川分野、道路分野とも)			
	整備者	各国道事務所、河川事務所			
	整備優先度	必要性	高くない	しやすさ	整備しづらい
	位置情報の形状	点(属性 position で記述)			
手順 (例)	<p>【初期整備(工事図面(紙媒体)より作成)】</p> <p>工事図面をスキャニング等によって、電子データにする。</p> <p>GIS上で、デジタイズ等によって管轄区間をデータ入力する。</p> <p>で作成したデータから、地理識別子を作成する。</p> <p>作成した地理識別子の品質評価を行う。</p> <p>作成した地理識別子を、地名辞典として地名辞典管理サービスに登録する。</p>				

また、「測点」の地理識別子の作成サンプル（XML 形式で作成した場合）を、表附 2-80 に示す。

表附 2-80 「測点」の地理識別子の作成サンプル

```
以下、場所インスタンスの1インスタンスの抜粋
<jci:PI_LocationInstance id="li_sp_XXXXXX">
  <jps:SI_LocationInstance.administrator>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName> 国道事務所
    </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001 </CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationInstance.administrator>
  <jps:SI_LocationInstance.position idref="pt_li_sp_XXXXXX" />
  <jci:PI_LocationInstance.geographicIdentifier>
    <jci:PI>
      <PI.value>
        <RdF>
          <Rd type="国道" name=" 国道事務所">
            <Rc name=" 工事">
              <Sp name=" " />
            </Rc>
          </Rd>
        </RdF>
      </PI.value>
      <PI.rs>road.srs.jacic.or.jp</PI.rs >
    </jci:PI>
  </jci:PI_LocationInstance.geographicIdentifier>
  <jci:PI_LocationInstance.gazetteer idref="gaz_sp" />
  <jci:PI_LocationInstance.locationType idref="lt_sp" />
</jci:PI_LocationInstance>
<jps:GM_Point id="pt_li_sp_XXXXXX">
  <jps:GM_Point.position>
    <jps:DirectPosition>
      <DirectPosition.coordinate>35.XXXXX 139. XXXXX</DirectPosition.coordinate>
      <DirectPosition.dimension>2</DirectPosition.dimension>
      <DirectPosition.coordinateReferenceSystem idref="rs001" />
    </jps:DirectPosition>
  </jps:GM_Point.position>
</jps:GM_Point>
```

(I) 「事務所管轄区域」の地理識別子の整備

「事務所管轄区域」の地理識別子の既存リソースとしては、河川分野・道路分野ともに「管内図」が考えられる。ただし、管内図は紙媒体であることがほとんどのため、整備にあたっては、スキャニングやデジタイズ等の作業が発生する。

また、地理識別子に付与する位置情報の形状は、「面」とするのがよい。

表附 2-81 に、「事務所管轄区域」の地理識別子の整備方法の概要と流れを示す。

表附 2-81 「事務所管轄区域」の地理識別子の整備の概要

概要	整備する地理識別子	事務所管轄区域			
	原典資料	管内図(河川分野、道路分野とも)			
	整備者	各国道事務所、河川事務所			
	整備優先度	必要性	高くない	しやすさ	整備しづらい
	位置情報の形状	面(属性 geographicExtent で記述)			
手順 (例)	<p>【初期整備(管内図(紙媒体)より作成)】</p> <p>管内図をスキャニング等によって、電子データにする。</p> <p>GIS 上で、デジタイズ等によって管轄区間をデータ入力する。</p> <p>で作成したデータから、地理識別子を作成する。</p> <p>作成した地理識別子の品質評価を行う。</p> <p>作成した地理識別子を、地名辞典として地名辞典管理サービスに登録する。</p>				

また、「事務所管轄区域」の地理識別子の作成サンプル（XML形式で作成した場合）を、表附 2-82 に示す。

表附 2-82 「事務所管轄区域」の地理識別子の作成サンプル

以下、場所インスタンスの1インスタンスの抜粋

```
<jci:PI_LocationInstance id="li_ofd_XXXXXX">
  <jps:SI_LocationInstance.administrator>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName> 国道事務所
    </CI_ResponsibleParty.individualName>
      <CI_ResponsibleParty.role>001 </CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationInstance.administrator>
  <jci:PI_LocationInstance.geographicIdentifier>
    <jci:PI>
      <PI.value>
        <Ofd name=" 国道事務所"/>
      </PI.value>
      <PI.rs>road.srs.jacic.or.jp</PI.rs >
    </jci:PI>
  </jci:PI_LocationInstance.geographicIdentifier>
  <jci:PI_LocationInstance.geographicExtent idref="exgb_ofd_XXXXXX" />
  <jci:PI_LocationInstance.gazetteer idref="gaz_ofd" />
  <jci:PI_LocationInstance.locationType idref="lt_ofd" />
</jci:PI_LocationInstance>
<jps:EX_GeographicBoundingBox id="exgb_ofd_XXXXXX">
  <jps:EX_GeographicBoundingBox.westBoundLongitude>140.XXXXX
</jps:EX_GeographicBoundingBox.westBoundLongitude>
  <jps:EX_GeographicBoundingBox.eastBoundLongitude>140.YYYYY
</jps:EX_GeographicBoundingBox.eastBoundLongitude>
  <jps:EX_GeographicBoundingBox.southBoundLongitude>36.XXXXX
</jps:EX_GeographicBoundingBox.southBoundLongitude>
  <jps:EX_GeographicBoundingBox.northBoundLongitude>36.YYYYY
</jps:EX_GeographicBoundingBox.northBoundLongitude>
</jps:EX_GeographicBoundingBox>
```

(J) 「出張所管轄区間」の地理識別子の整備

「出張所管轄区間」の地理識別子の既存リソースとしては、河川分野・道路分野ともに「管内図」が考えられる。ただし、管内図は紙媒体であることがほとんどのため、整備にあたっては、スキャニングやデジタイズ等の作業が発生する。

また、地理識別子に付与する位置情報の形状は、「面」とするのがよい。

表附 2-83 に、「出張所管轄区間」の地理識別子の整備方法の概要と流れを示す。

表附 2-83 「出張所管轄区間」の地理識別子の整備の概要

概要	整備する地理識別子	出張所管轄区間			
	原典資料	管内図(河川分野、道路分野とも)			
	整備者	各国道事務所、河川事務所			
	整備優先度	必要性	高くない	しやすさ	整備しづらい
	位置情報の形状	面(属性 geographicExtent で記述)			
手順 (例)	<p>【初期整備(管内図(紙媒体)より作成)】</p> <p>管内図をスキャニング等によって、電子データにする。</p> <p>GIS 上で、デジタイズ等によって管轄区間をデータ入力する。</p> <p>で作成したデータから、地理識別子を作成する。</p> <p>作成した地理識別子の品質評価を行う。</p> <p>作成した地理識別子を、地名辞典として地名辞典管理サービスに登録する。</p>				

また、「出張所管轄区間」の地理識別子の作成サンプル（XML 形式で作成した場合）を、表附 2-84 に示す。

表附 2-84 「出張所管轄区間」の地理識別子の作成サンプル

以下、場所インスタンスの 1 インスタンスの抜粋

```
<jci:PI_LocationInstance id="li_bofs_XXXXXX">
  <jps:SI_LocationInstance.administrator>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName> 国道事務所
    </CI_ResponsibleParty.individualName>
    <CI_ResponsibleParty.role>001 </CI_ResponsibleParty.role>
    </jps:CI_ResponsibleParty>
  </jps:SI_LocationInstance.administrator>
  <jci:PI_LocationInstance.geographicIdentifier>
    <jci:PI>
      <PI.value>
        <BOFS name=" 出張所管轄区間"/>
      </PI.value>
      <PI.rs>road.srs.jacic.or.jp</PI.rs >
    </jci:PI>
  </jci:PI_LocationInstance.geographicIdentifier>
  <jci:PI_LocationInstance.geographicExtent idref="exgb_bofs_XXXXXX" />
  <jci:PI_LocationInstance.gazetteer idref="gaz_bofs" />
  <jci:PI_LocationInstance.locationType idref="lt_bofs" />
</jci:PI_LocationInstance>
<jps:EX_GeographicBoundingBox id="exgb_bofs_XXXXXX">
  <jps:EX_GeographicBoundingBox.westBoundLongitude>140.XXXXX
</jps:EX_GeographicBoundingBox.westBoundLongitude>
  <jps:EX_GeographicBoundingBox.eastBoundLongitude>140.YYYYY
</jps:EX_GeographicBoundingBox.eastBoundLongitude>
  <jps:EX_GeographicBoundingBox.southBoundLongitude>36.XXXXX
</jps:EX_GeographicBoundingBox.southBoundLongitude>
  <jps:EX_GeographicBoundingBox.northBoundLongitude>36.YYYYY
</jps:EX_GeographicBoundingBox.northBoundLongitude>
</jps:EX_GeographicBoundingBox>
```

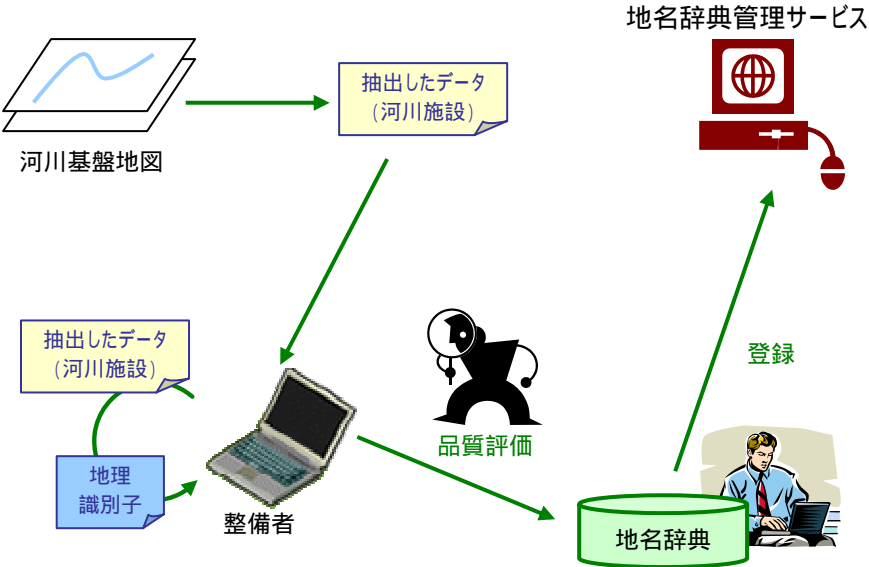
(K) 「施設」の地理識別子の整備

「施設」の地理識別子の既存リソースとしては、河川分野では「河川基盤地図」、道路分野では「DRM」など考えられ、この中から既存リソースを収集し、地理識別子を作成する。

また、地理識別子に付与する位置情報の形状は、「点」とするのが最も簡単に作成ができる。

表附 2-85 に、「施設」の地理識別子の整備方法の概要と流れを示す。

表附 2-85 「施設」の地理識別子の整備の概要

概要	整備する地理識別子	施設			
	原典資料	河川基盤地図(河川分野の場合) DRM(道路分野の場合)			
	整備者	各国道事務所、河川事務所			
	整備優先度	必要性	高い	しやすさ	整備しやすい
	位置情報の形状	点(属性 position で記述)			
手順	<p>【初期整備(河川基盤地図を既存リソースとする場合)】</p> <p>河川基盤地図から、河川施設を示すデータ及び対応する位置情報のデータを抽出する。</p> <p>で抽出したデータを用いて、地理識別子を作成する。</p> <p>作成した地理識別子の品質評価を行う。</p> <p>作成した地理識別子を、地名辞典として地名辞典管理サービスに登録する。</p> 				

また、「施設」の地理識別子の作成サンプル（XML 形式で作成した場合）を、表附 2-86 に示す。

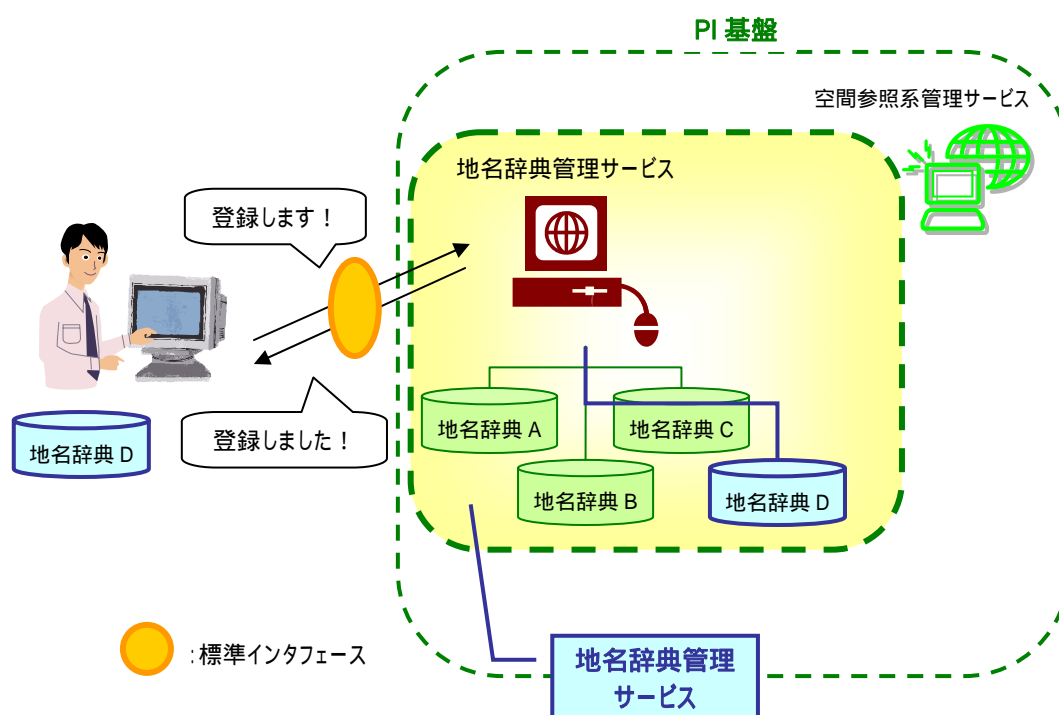
表附 2-86 「施設」の地理識別子の作成サンプル

```
以下、場所インスタンスの1インスタンスの抜粋
<jci:PI_LocationInstance id="li_fac_XXXXXX">
  <jps:SI_LocationInstance.administrator>
    <jps:CI_ResponsibleParty>
      <CI_ResponsibleParty.individualName> 河川事務所
    </CI_ResponsibleParty.individualName>
    <CI_ResponsibleParty.role>001 </CI_ResponsibleParty.role>
  </jps:CI_ResponsibleParty>
</jps:SI_LocationInstance.administrator>
<jps:SI_LocationInstance.position idref="pt_li_fac_XXXXXX" />
<jci:PI_LocationInstance.geographicIdentifier>
  <jci:PI>
    <PI.value>
      <Fac name="利根川 堤防"/>
    </PI.value>
    <PI.rs>river.srs.jacic.or.jp</PI.rs >
  </jci:PI>
</jci:PI_LocationInstance.geographicIdentifier>
<jci:PI_LocationInstance.gazetteer idref="gaz_fac" />
<jci:PI_LocationInstance.locationType idref="lt_fac" />
</jci:PI_LocationInstance>
<jps:GM_Point id="pt_li_fac_XXXXXX">
  <jps:GM_Point.position>
    <jps:DirectPosition>
      <DirectPosition.coordinate>35.XXXXX 139. XXXXX</DirectPosition.coordinate>
      <DirectPosition.dimension>2</DirectPosition.dimension>
      <DirectPosition.coordinateReferenceSystem idref="rs001" />
    </jps:DirectPosition>
  </jps:GM_Point.position>
</jps:GM_Point>
```

2-4. 地名辞典管理サービスへの地名辞典の登録

附属書 2-3 でも触れたが、作成した地名辞典データは、地名辞典管理サービスにデータを登録することで、利用することが可能となる。

地名辞典管理サービスについては、本編第 5 章でその概要を示したが、地名辞典管理サービスは、他のシステムとサービスやデータを相互に連携させるために、別途定める「[建設情報連携ポータル標準インタフェースガイドライン](#)」に示された標準インタフェースを実装しなければならない。



図附 2-22 地名辞典の登録

3. 地名辞典の運用方法

地名辞典は、初期整備だけでなく、その後どのように更新をしていくかも重要な問題である。そこで、地名辞典の更新方法についても解説するとともに、標準インタフェースを実装した地名辞典サービスを立ち上げた場合に可能な、サービスを利用した地理識別子の登録・修正方法を解説する。また、標準インタフェースを実装していない場合の地理識別子の登録、削除、修正に際し注意すべき点を、本編で示した規約を引用して解説する。

地名辞典のデータは、GISにおける背景地図データと同様に、データの新鮮さを保つことが重要である。たとえば、新しく建設された施設名が地名辞典に登録されないままであったり、市町村合併等によって新しい住所が制定されてもその情報が地名辞典に登録されていない場合は、利用価値の高い地名辞典とはいえない。

本附属書では、地名辞典の運用方法について、二つの方法を紹介する。一つは、地理識別子を定期的にメンテナンスし、情報を更新する方法である。更新量が多い地理識別子では手動で更新作業を行うことは難しいため、定期的にメンテナンスを行うことが必要である。この場面では既存リソースの更新のタイミング等が重要になるため、適切な地理識別子の更新方法を解説する。

二つ目は、日常的に修正のあった地理識別子の情報を随時修正していく方法である。この方法では、一つ目の方法のようにある周期で定期的にメンテナンスを掛けるのではなく、例えば名称が変更になった施設や、新たに建設された施設の地理識別子を随時手動で更新を行う。標準インタフェースガイドラインにしたがって地名辞典管理サービスを実装し、地理識別子を登録・修正する機能をもっている場合は、地名辞典管理サービスを利用して地理識別子を登録・修正することができる。地名辞典管理サービスを実装していない場合は、地名辞典のデータベースを直接編集し、地理識別子を登録・修正する

また、地理識別子は、古い地理識別子もデータとして残しておくことが重要な場合がある。例えば、以前の図面等に記載された古い住所や施設名をキーワードとして検索を掛ける場面も少なくない。そのような場合のために、新しい地名辞典を作成した場合は、データを入れ換えるのではなく、追加をしていくことが重要である。

3-1. 地名辞典の更新

3-1-1. 定期的に地名辞典を更新する方法

地名辞典の更新方法は、地理識別子によって原典資料ごとに更新のタイミングが異なるために様々である。そこで、この項目では地理識別子ごとに、適切な更新方法の手順を解説する。また、更新に際しては本編第3章に示す「地名辞典の整備・運用のための規約」

に則っている必要がある。

(A) 住所

< 更新のタイミング >

住所は、街区レベル位置参照情報をはじめ既に整備されたリソースがあるので、そのリソースの更新に合わせて住所の地名辞典を更新するのが良い。

表附 3-1 住所の既存リソースと、その更新のタイミング

既存リソース		既存リソースの更新時期
1	街区レベル位置参照情報	1回/年
2	数値地図 2500/25000	不定期
3	民間データ	採用した民間データにより異なる

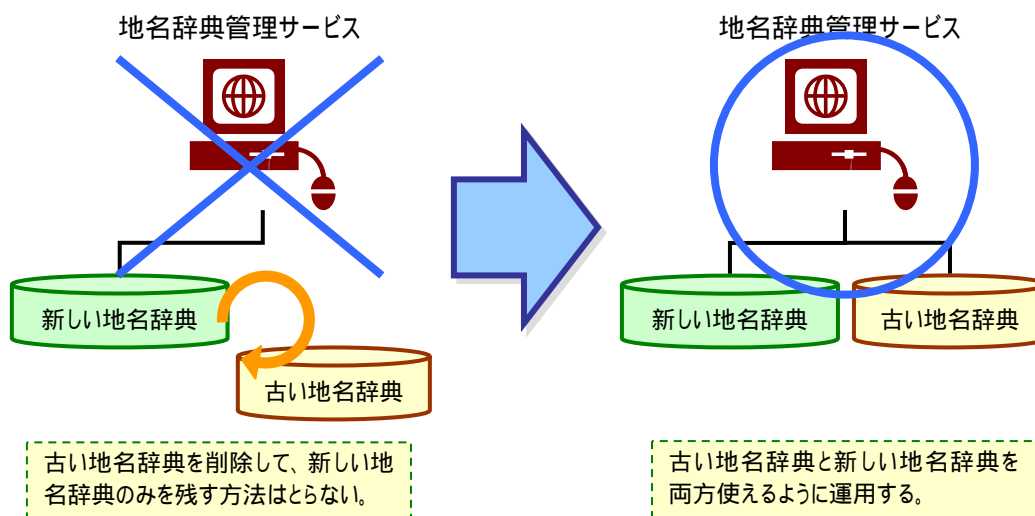
< 更新作業の実施者 >

定期的に更新される既存リソースを使用して、整備局等が更新した地名辞典を作成し、配布するのが良い。

< 更新時の考え方 >

住所の地名辞典は、更新前の住所の地名辞典を残しつつ新規に整備した地名辞典を追加していき、古い辞典と新しい辞典の両方を利用できるように運用することが重要である。

例えば、電子納品された情報に記載の住所が後に変更になった場合、最新の住所の地名辞典のみを残す運用だと、旧住所を入力して検索をかけてもその情報に辿りつくことができない。古い地名辞典と最新の地名辞典をともに利用できるようにしておくことで、旧住所にも対応した住所の地名辞典とすることができる。



図附 3-1 地名辞典の更新時の考え方

(B) 河川、河川施設、水系、流域、河川距離標

< 更新のタイミング >

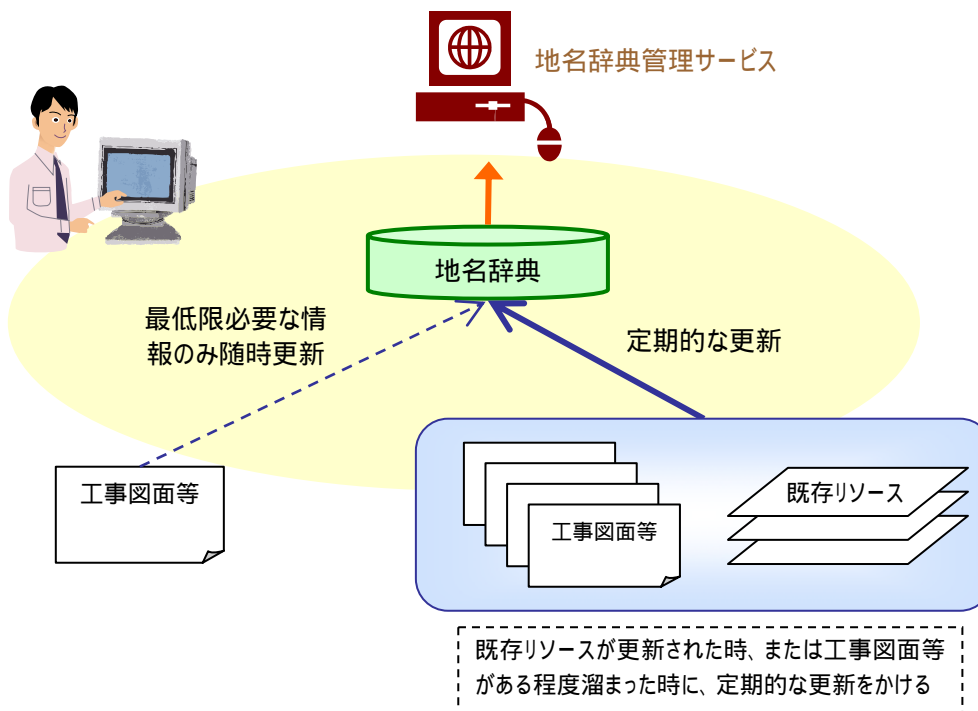
既存リソースとなる河川基盤地図や国土数値情報は、住所における街区レベル位置参照情報のような定期的な更新時期をもたず、また、数年単位の更新の可能性も低いと考えられる。

河川、河川施設、水系、流域、河川距離標の定期的な更新は、
 既存リソースとなる河川基盤地図や国土数値情報が更新された時
 河川の改修工事等が行われた際の工事用図面等がある程度溜まった時
 に実施するのがよい。

上記の更新時期が来るまでの間は、工事用図面等を用いて、必要最小限の情報のみを地理識別子として追加しておくのがよい。

表附 3-2 河川分野の既存リソースと、その更新のタイミング

既存リソース		既存リソースの更新時期
1	河川基盤地図/河川図(2500)	不定期
2	河川基盤地図/流域図(25000)	不定期
3	国土数値情報	不定期



図附 3-2 河川、河川施設、水系、流域、河川距離標の定期的な更新の考え方

< 更新作業の実施者 >

その河川を管理する河川事務所が更新を行い、地名辞典を更新するのがよい。

(C) 管理路線、道路施設、道路距離標、交差点

< 更新のタイミング >

既存リソースとなる DRM は、住所における街区レベル位置参照情報のような定期的な更新時期をもたず、また、数年単位の更新の可能性も低いと考えられる。また、もう一つの既存リソースとなる「道路管理システム」のデータも、随時追加がされるが、定期的にデータのメンテナンスを行うことはあまりない。

したがって、管理路線、道路施設、道路距離標、交差点の定期的な更新は、

DRM を既存リソースとする場合は、DRM が更新された時

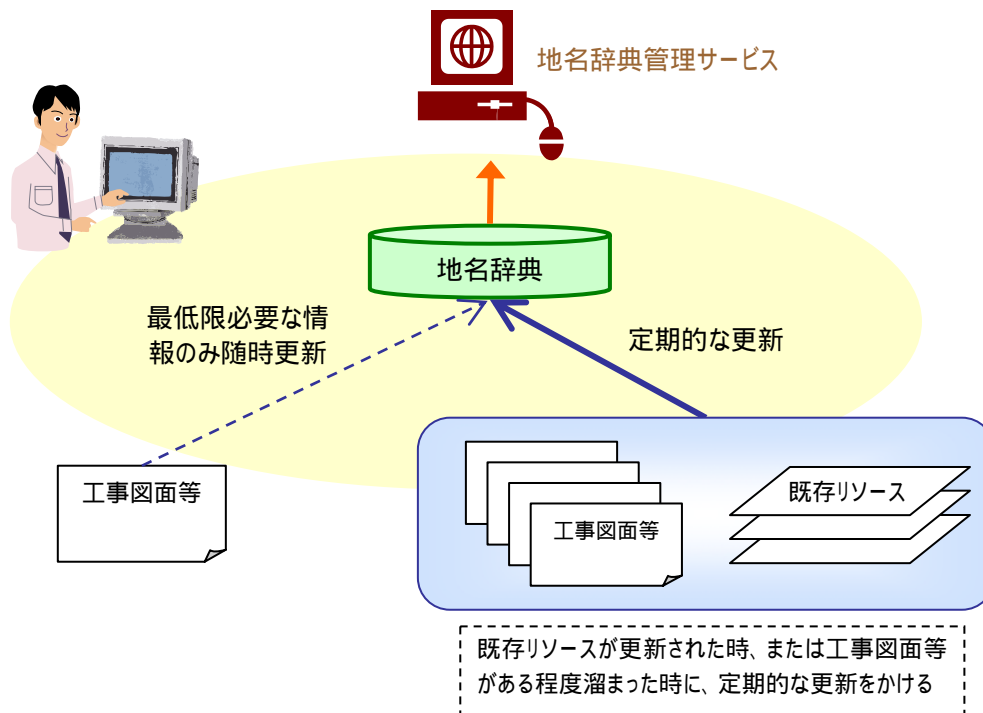
「道路管理システムデータ」を既存リソースとする場合は、改修工事等が行われた際の工事用図面等がある程度溜まった時

に実施するのがよい。

上記の更新時期が来るまでの間は、工事用図面等を用いて、必要最小限の情報のみを地理識別子として追加しておくのがよい。

表附 3-3 管理路線、道路施設、道路距離標、交差点の既存リソースと、その更新のタイミング

既存リソース		既存リソースの更新時期
1	DRM	不定期
2	「道路管理システム」データ	不定期



図附 3-3 管理路線、道路施設、道路距離標、交差点の定期的な更新の考え方

< 更新作業の実施者 >

その道路を管理する国道事務所が更新を行い、地名辞典を更新するのがよい。

(D) 事務所管轄区域、出張所管轄区間

< 更新のタイミング >

既存リソースとなる管内図は、定期的な更新時期をもたず、また、数年単位の更新の可能性も低いと考えられる。管内図が更新されても、事務所管轄区域と出張所管轄区間の地理識別子の更新が必要な場合は管轄区域や管轄区間が変更になった場合のみであり、ごく稀なケースである。

また、事務所管轄区域と出張所管轄区間は、作成する地理識別子は、その区域及び区間のみであり、作業量はごく僅かである。

したがって、事務所管轄区域と出張所管轄区間については定期的な更新は行わず、管轄区域や管轄区間が変更になり、管内図が更新された時点で、更新を行えばよい。

< 更新作業の実施者 >

その管轄区域や管轄区間を有する事務所が更新を行い、地名辞典を更新するのがよい。

3-1-2. 随時地名辞典を更新する方法

既存リソースの更新のタイミングに合わせた地名辞典の更新では業務に支障をきたす場合や、地理識別子の更新量が小さい場合は、作成した地理識別子の内容に誤り（名称や地理範囲等）があった場合は、随時更新を行うと、地名辞典の情報の鮮度を保つことができ、業務での利用においても非常に有効である。

住所や新たに道路が建設された場合の道路距離標などは、全ての住所表記や距離標を追加するのは作業量が非常に多くなるが、基本的な項目（住所における「 町 1 丁目」や距離標における「 kp」等）のみを手動で追加しておくだけでも、業務での利用において非常に有効である。（この場合、残りの細かい項目（住所における「 町 1 丁目 1-1」や距離標における「 . x kp」等）は、定期更新の際に整備すればよい）

地名辞典を随時更新する方法は、標準インタフェースを実装した地名辞典管理サービスを立ち上げている場合とそうでない場合で、方法が異なる。

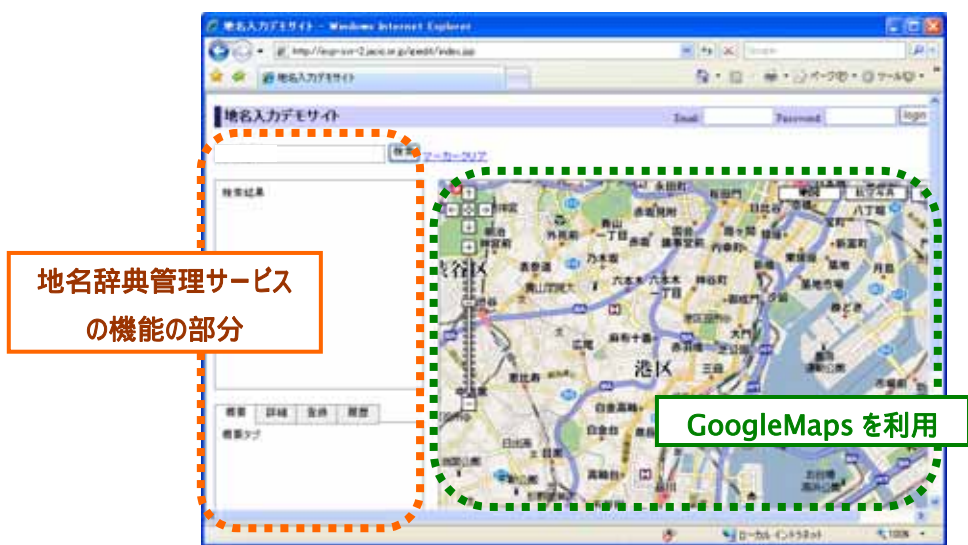
標準インタフェースを実装した地名辞典管理サービスを立ち上げている場合は、サービスを利用して地理識別子の登録、削除、更新を画面上で行うことができる。

一方、標準インタフェースを実装した地名辞典管理サービスを立ち上げていない場合は、地名辞典に格納したデータに直接編集をかけ、地理識別子の追加、削除、更新を行う。この場合、本編で定める規約にしたがって追加・更新・削除を行わなければならない、注意が必要である。

(A) 標準インタフェースを実装している場合

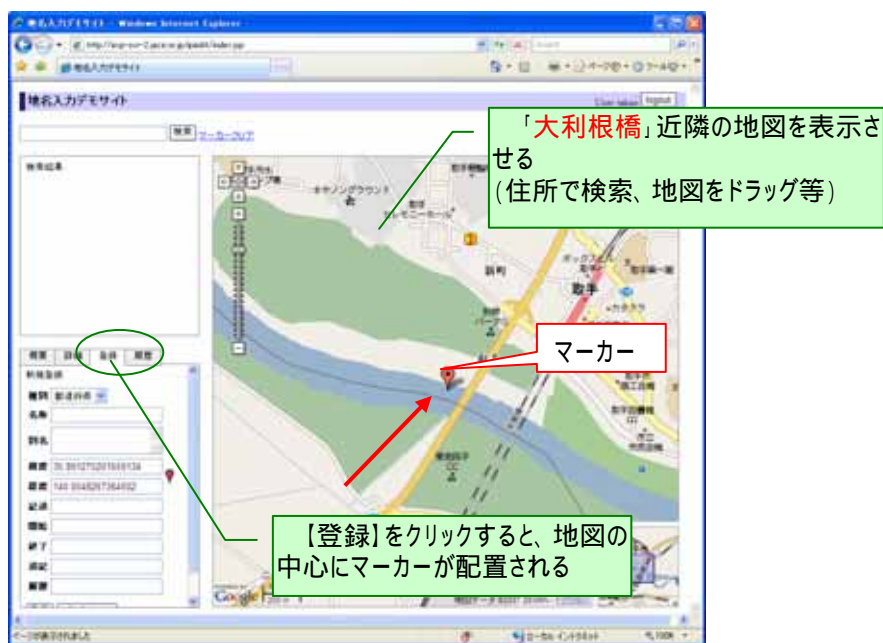
標準インタフェースを実装した地名辞典管理サービスを立ち上げている場合は、標準インタフェースに含まれる登録・削除機能を利用して、地理識別子の登録、削除を行うことも可能となる。また、もし整備した地名辞典の情報に誤りがあった場合は、標準インタフェースに含まれる更新機能を利用して、地理識別子の情報を修正することができる。

以下に、標準インタフェースを実装した地名辞典管理サービスを立ち上げている場合の、地理識別子の登録、削除、修正の例を示す。なお、この例で紹介する地名辞典管理サービスでは、背景地図に GoogleMaps を利用し、GoogleMaps の地図上で位置を指定した登録を可能とする機能をもっている。



図附 3-4 地名辞典管理サービスの画面例

【新規に道路施設(大利根橋)を追加する例】



図附 3-5 新規に道路施設を追加する例



図附 3-6 新規に道路施設を追加する例



図附 3-7 新規に道路施設を追加する例

(B) 標準インターフェースを実装していない場合

標準インターフェースを実装した地名辞典管理サービスを立ち上げていない場合は、地名辞典に格納したデータに直接編集をかけ、地理識別子の追加、削除、更新を行う。この場合、本編で定める規約にしたがって追加・更新・削除を行わなければならない、注意が必要である。

4. 地名辞典の拡充

地名辞典を運用していく中で、整備する地理識別子の種類を追加し、地名辞典を拡充することで、より利用価値の高い地名辞典としていく場面が出てくる。そのような場合は、附属書 2-1 で検討した整備する地理識別子の優先度選定結果を役立てることができる。もし、優先度設定の際に候補として挙がっており、まだ整備していない地理識別子があれば、その地理識別子を整備することで、より多くの場面で地名辞典を利用することができるようになる。

一方、前回の地理識別子の優先度選定検討から時間が経過している場合や、新たな要素を加味した地理識別子の選定を行う場合は、附属書 2-1 で示した指針に基づき、再度拡充する地理識別子の選定作業を実施するとよい。

以下に、想定される地名辞典の拡充の流れごとに、考えるべき点を示す。

追加する地理識別子の種類を決定する

- ✓ 以前の優先度設定の結果を利用する場合
 前回の結果から優先度の高かったものを整備する。
- ✓ 以前の優先度設定の結果を利用しない場合
 優先度設定をもう一度行い、新たに追加する地理識別子の種類を決定する。

追加する地理識別子の場所型を作成する

- ✓ 本編第 3 章及び附属書 2-2 を参考に、新たに場所型を追加する。
- ✓ 他の機関などが作成した既存の場所型を利用できる時は、それを利用する。

地理識別子による空間参照系の変更または新規作成を検討する

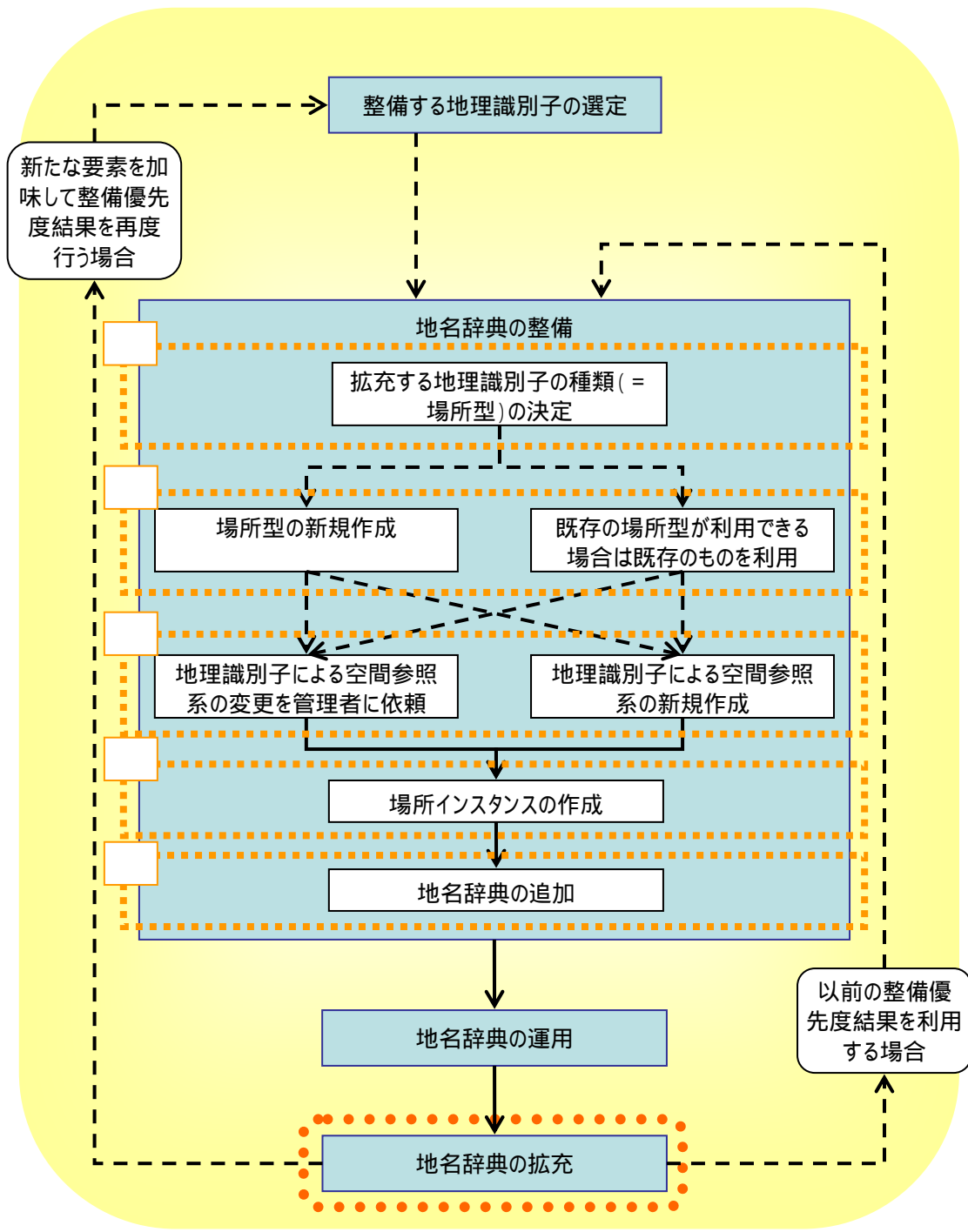
- ✓ によって、地理識別子による空間参照系を変更する必要がある場合は、地理識別子による空間参照系の管理者に、変更を依頼する。
- ✓ 新たに地理識別子による空間参照系を作成する場合は、本編第 3 章及び附属書 2-2 を参考に、新たに場所型を追加する。

地理識別子を含む場所インスタンスと地名辞典を作成する

- ✓ 本編第 3 章及び附属書 2-2 を参考に、場所型にある地理識別子の記述方法に基づいて場所インスタンスを作成し、それを新規に作成した地名辞典に格納する。

地名辞典を追加する

- ✓ で作成した地名辞典を、地名辞典管理サービスに追加する。



図附 4-1 地名辞典の拡充の流れ

地名辞典は一度整備したら終わり、ということではなく、運用をしていながら随時情報を更新し、かつ業務での利用上必要となる地理識別子を追加していくことで、常に地名辞典として進化していくとともに、より使いやすい地名辞典となる。

CAD-GIS 連携の手引き書（案）

第 1 版

平成 19 年 11 月

国土交通省

はじめに

公共事業で扱う情報は、属地性が高いことから、多様な情報を電子地図上で利用できる GIS の活用が各機関で推進されている。また、地理情報標準プロファイル (JPGIS) 形式に則したデータの整備も推進されてきている。測量、調査、設計および施工では、CAD データが広く利用されており、維持管理では、GIS による構造物の管理の整備が推進されている。ライフサイクルの各段階の業務効率化を実現するためには、各フェーズ間の情報の連携が重要であり、各フェーズで扱うデータの特性を踏まえて、鮮度の高いデータを整備して構造物を管理するには、工事完成図の CAD データを利用した GIS データの整備が有効な方策の一つである。

このような認識の下、国土交通省では道路工事を対象に、CAD ソフトで作成した完成平面図に格納された各種データを GIS で利用するための「道路工事完成図等作成要領」を策定し、試行を経て平成 18 年 8 月に施行している。同要領は、CAD データを GIS データに変換するための CAD ソフトによる完成平面図の作成方法を定めたものである。

GIS のデータ構造は、各分野によって定義が異なっており、そのため公共事業の各分野で適用できる CAD から GIS の変換・連携の汎用的な作成方法を定めた指針を作成するのは非常に困難であることから、本書は、「道路工事完成図等作成要領」と同じように、各分野において維持管理段階で利用する GIS アプリケーションの GIS データ(基盤地図データ)に、測量、設計、施工段階等の建設情報のライフサイクルの前段階で作成される CAD データを有効利用しようとした時に、CAD-GIS 間のデータ変換を進めるにあたり必要となる検討事項や考え方をとりまとめたものである。

本書の策定にあたっては、国土交通省が(財)日本建設情報総合センターに業務を委託し、建設情報標準化委員会 電子地図/建設情報連携小委員会(柴崎亮介小委員長)に、ご検討・ご審議をいただいた。

策定に当たって多大なご協力をいただいた関係各位に深く謝意を表する次第である。

平成 19 年 11 月
国土交通省

目 次

1. CAD GIS連携の手引書の概要.....	1
1.1 目的.....	1
1.2 適用範囲.....	2
1.3 関連する基準類.....	3
2. CADとGISについて.....	4
2.1 完成平面図データ（CADデータ）について.....	4
2.2 基盤地図データ（GISデータ）について.....	5
2.3 CAD-GIS連携の課題.....	5
2.4 必要となる（作成すべき）基準類等.....	6
3. CAD-GIS連携の手順.....	7
4. データ変換にあたっての基本的規定事項.....	9
4.1 完成図データの作成について.....	9
4.1.1 ファイル形式.....	9
4.1.2 座標系.....	10
4.1.3 作成する図形.....	11
4.1.4 レイヤ分類.....	13
4.1.5 属性入力.....	17
4.1.6 点データの作成.....	18
4.1.7 線データの作成.....	19
4.1.8 面データの作成.....	21
4.2 GISデータへの変換について.....	23
4.2.1 データ変換ツール.....	23
4.2.2 データ変換ツールの基本的な機能.....	23

1. CAD GIS 連携の手引書の概要

1.1 目的

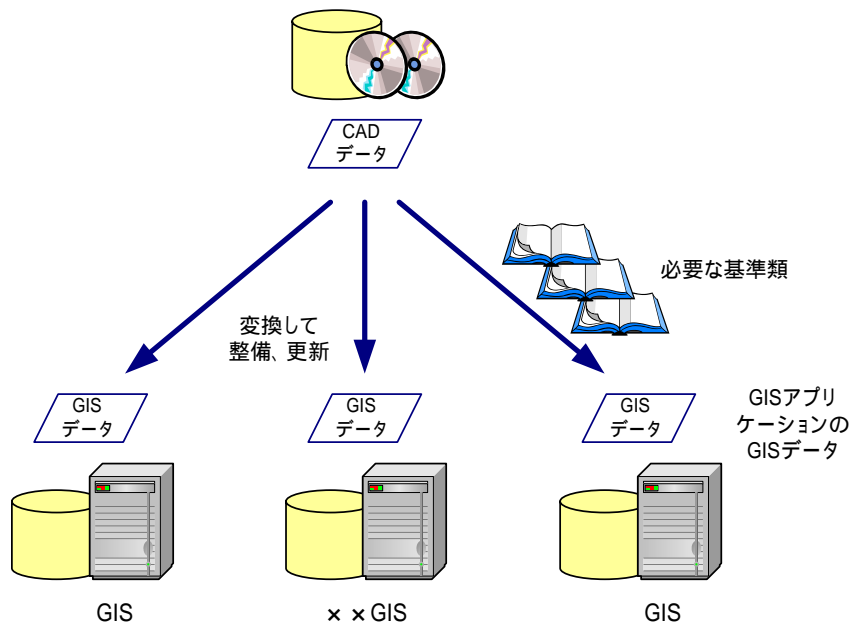
本手引き書は、CAD データを利用して、地理情報標準に準拠した GIS データを整備、更新していくための運用に必要なとなる基準類等の作成を検討する際の考え方をとりまとめたものである。

【解説】

本手引き書は、維持管理段階で利用する GIS アプリケーションの GIS データ（基盤地図データ）に、測量、設計、施工段階等の建設情報のライフサイクルの前段階で作成される CAD データを有効利用したいと考えている発注者が、CAD-GIS 間のデータ変換を進めるにあたり必要となる基準類等の作成を検討する際の考え方をとりまとめたものである。

本手引き書に準じて必要な基準類等を作成することより、CAD データから GIS データへの変換作成の実現を目指すことを目的とする。

なお、本手引き書は、データ変換に係る汎用的な内容を示すものであり、様々な分野に適用できるものである。ただし、国土交通省における道路工事完成図等作成要領の取組を参考にして定めたものであり、道路分野の施工段階で電子納品される CAD データ（完成平面図データ）の有効利用が基本となっている。



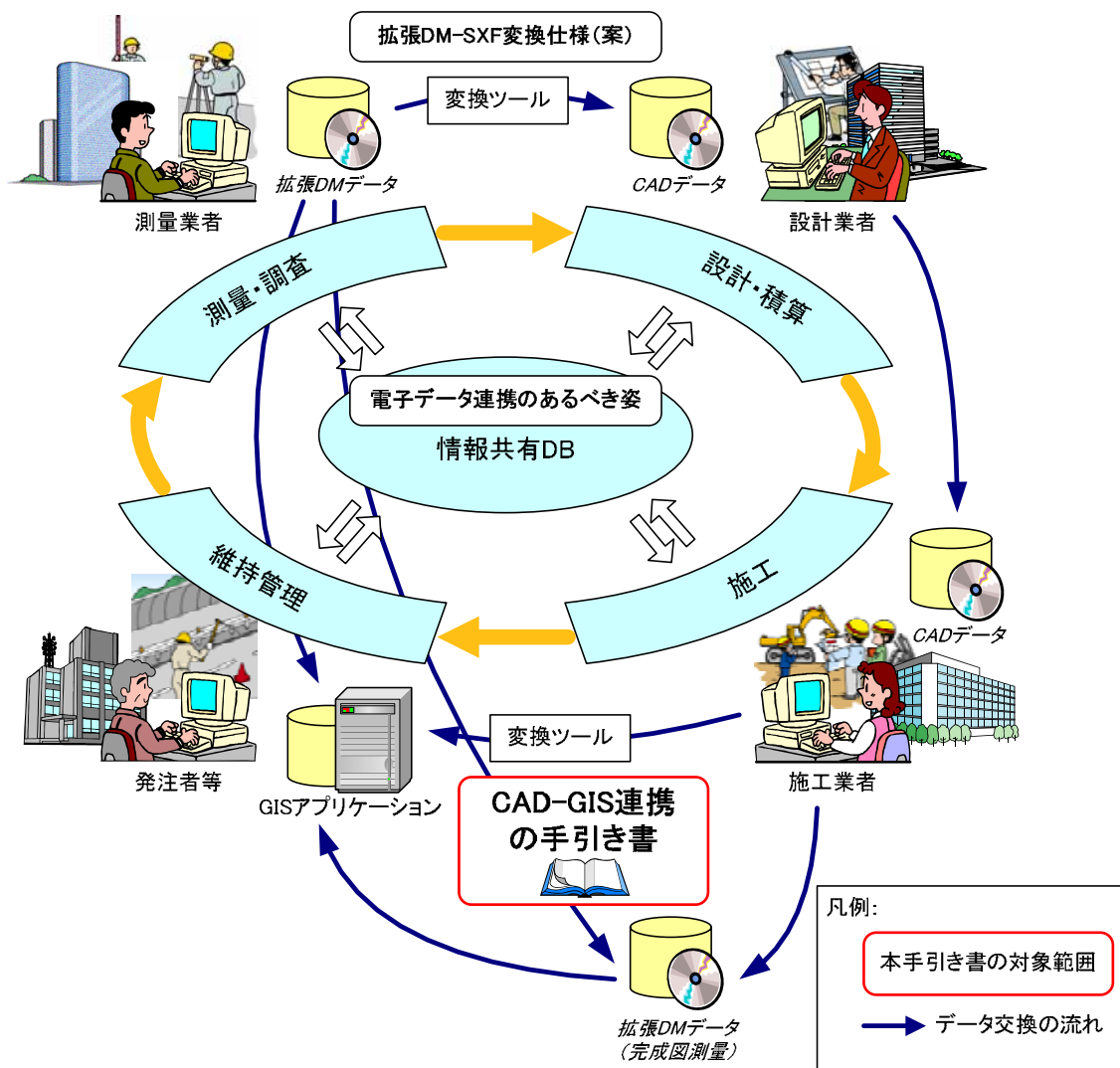
道路工事完成図等作成要領の詳細は、国土交通省国土技術政策総合研究所の「道路工事完成図等作成支援サイト（<http://www.nilim-cdrw.jp/index.html>）」を参照して下さい。

1.2 適用範囲

本手引き書は、DM、CAD、GIS 間のデータ連携の全体像の中で、基本的に施工業者（施工の CAD データ）から発注者等（維持管理の GIS アプリケーション）までの範囲を対象とする。

【解説】

DM、CAD、GIS 間のデータ連携・達成イメージにおける本手引き書の位置づけは、下図のとおりであり、施工業者（施工の CAD データ）から発注者（維持管理の GIS アプリケーション）までを対象とする。



1.3 関連する基準

本手引き書に関連する要領・基準類は、以下のとおりである。

- (1) 電子納品
 - 「CAD 製図基準 (案)」
 - 「工事完成図書の電子納品要領 (案)」
- (2) CAD データ
 - 「SXF Ver.3.1 仕様書・同解説」
- (3) GIS データ
 - 「地理情報標準プロファイル (JPGIS) Ver.1.0」

【解説】

- (1) 電子納品に関する要領・基準類
 - 1) 「CAD 製図基準 (案)」
(平成 16 年 6 月 国土交通省)
 - 2) 「工事完成図書の電子納品要領 (案)」
(平成 16 年 6 月 国土交通省)
- (2) CAD データに関する要領・基準類
 - 1) 「SXF Ver.3.1 仕様書・同解説」
(平成 19 年 11 月 国土交通省)
- (3) GIS データに関する要領・基準類
 - 1) 「地理情報標準プロファイル (JPGIS) Ver.1.0」
(平成 17 年 1 月 国土交通省国土地理院)

2. CAD と GIS について

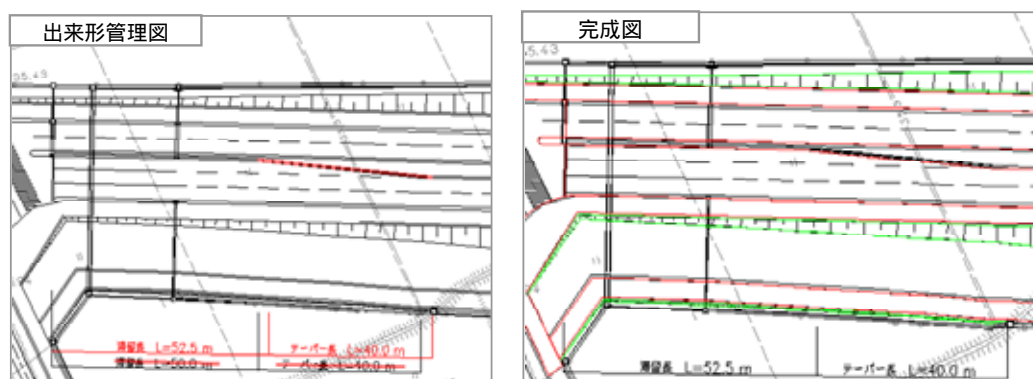
2.1 完成平面図データ (CAD データ) について

工事の契約図書に基づいて作成される完成平面図の CAD データを工事後の維持管理段階における GIS データ (基盤地図データ) の整備、更新に有効利用する。

【解説】

工事の契約図書 (「土木工事共通仕様書」) の中では、「完成図」の提出を規定しているが、完成図 = 出来形管理図として設計値に対して出来形値を () 書きで対応づけ、出来形形状を色分けして示したり、完成図 = 出来形管理図として当初設計値や形状を消去して完成形状のみを示した図面を作成したり、「完成図」の定義は、運用上明確になっていない。

「出来形管理図」は、「土木工事施工管理基準及び規格値 (案)」の中で定められ、請負者が施工管理を行う上で必要となる、設計値と施工された出来形との差異を示すものであり、「完成図」を“施工された公物の管理及び以後の工事の計画・設計等に利用することを目的に工事目的物の完成形状を示した図面”として定義し、「出来形管理図」と区別する。



完成図には、平面図、縦断図、横断図、構造図、構造詳細図などがあるが、工事後の維持管理段階における管理図や次回工事発注図等として利用される完成平面図の CAD データを GIS データ (基盤地図データ) の整備、更新に有効利用する。

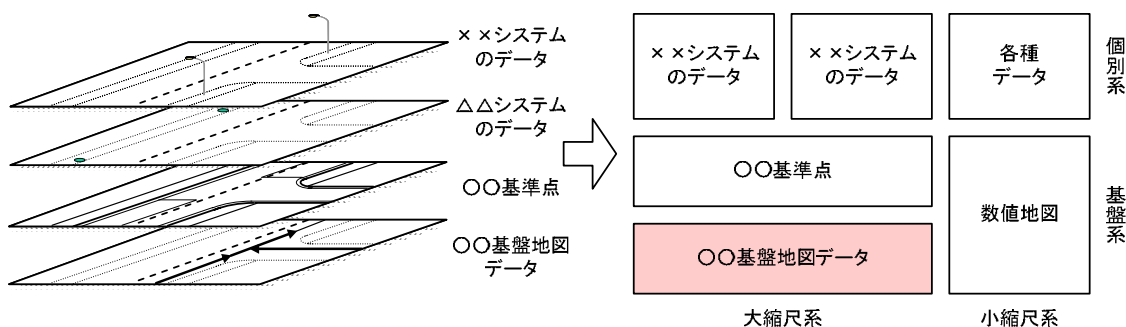
2.2 基盤地図データ（GIS データ）について

施工段階で電子納品される完成平面図の CAD データを利用して、維持管理段階で利用する GIS の空間データのうち基盤地図データを整備、更新する。

【解説】

維持管理段階で利用する「GIS」は、空間データとそれらを利用したアプリケーション・システムを包括し、情報基盤となる空間データ（基盤系）及び既存のデータベース・システム（個別系）により、構成される。

このうち、「基盤地図データ」は、既存のデータベース・システムとの役割分担の下、基盤として構築すべき最小限の共有空間データとして位置づけられ、“GIS で用いる空間データのうち、各種サービスを実現する上で必要となる共用性の高いデータ”と定義する。つまり、GIS では、基盤地図データと既存のデータベース・システム間での連動（リンク）を確保し、各データの特徴を活かした組合せによるシステム利用を図る。



GIS は、様々な GIS データにより構成されるが、電子納品される完成平面図の CAD データを利用して整備、更新されるのは、基盤地図データのみである。

2.3 CAD-GIS 連携の課題

GIS は、利用用途によって、対象となる地物、必要となる属性情報も異なることから、GIS データの製品仕様及びそれに基づいた CAD-GIS の変換仕様を個々に作成する必要がある。

【解説】

GIS は、利用用途によって、対象となる地物、必要となる属性情報も異なり、基準となる標準製品仕様がない。また、GIS（空間データ）の製品仕様は、地理情報標準プロファイル（JPGIS）に準拠していないものも多い。

2.4 必要となる（作成すべき）基準類等

CAD データを利用して、GIS データを整備、更新していくためには、以下のような基準類等を新たに作成する必要がある。

- (1) 完成図の作成要領
- (2) データ交換の仕様
- (3) データ変換のツール
- (4) GIS データの製品仕様書

【解説】

(1) 完成図の作成要領

GIS データに変換可能な完成図データの作成方法を記載したマニュアルを作成する必要がある。

(例)「道路工事完成図等作成要領」(国土交通省)

(2) データ交換の仕様

完成図データ(CAD データ)で、図形に属性を与える仕組みや取扱いを定めた規約(属性セット)を作成する必要がある。

(例)「道路基盤データ交換属性セット(案)」(国土交通省)

(3) データ変換のツール

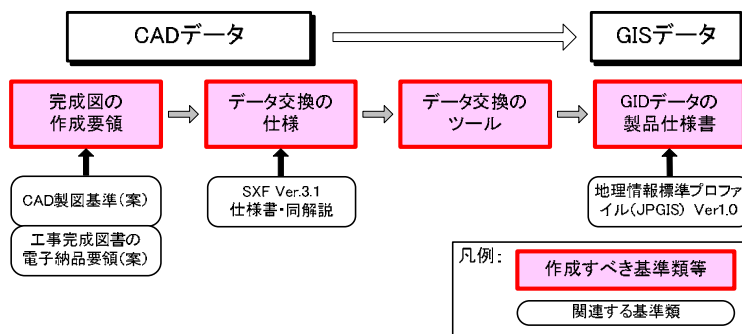
完成図データ(CAD データ)をGIS データ(JPGIS 対応 XML)に変換するコンバータを構築する必要がある。

(例)「CAD-GIS コンバータ【道路版】」(国土交通省)

(4) GIS データの製品仕様書

JPGIS に準拠した GIS データの製品仕様書を作成する必要がある。

(例)「道路基盤データ製品仕様書(案)」(国土交通省)



道路工事完成図等作成要領は、SXF Ver.3.0 に基づき作成されているが、CAD-GIS連携の手引書(案)はSXF Ver3.1 を前提に作成されている。SXF Ver.3.1 はクロソイドフィーチャー、弧長寸法線フィーチャーの追加と、属性付加機構が改訂されている。

3. CAD-GIS 連携の手順

CAD-GIS 連携は、以下の手順で進めるものとする。

- <手順 1> 「GIS データの製品仕様書」を作成する。
- <手順 2> 完成図で取得対象とする「地物項目」、「図形要素」を規定する。
- <手順 3> 地物の「レイヤ分類方法」を規定する。
- <手順 4> 各地物に与える「属性」を規定する。
- <手順 5> 「完成図の作成要領」及び「データ交換仕様」を作成する。
- <手順 6> CAD-GIS 間の「データ変換ツール」を作成する。

以上の手順で作成した要領、仕様書に準じて CAD で完成図を作成し、データ変換ツールによりデータ変換を行う。

【解説】

「2.3 CAD-GIS 連携の課題」で記したとおり、GIS は利用用途により、対象とする地物や必要となる属性情報が異なるため、完成図（CAD）からデータを取得して GIS へ受け渡す場合には、以下の項目について検討・規定する必要がある。

(1) 「GIS データの製品仕様書」の作成

維持管理で必要となる情報を精査し、GIS データの製品仕様書を作成する。

(2) 「地物項目」、「図形要素」の規定

完成図で取得し、GIS へと受け渡す「地物項目（図形名称）」及び「図形要素（図形の種類）」（点、線、面）を規定する。

(3) 「レイヤ分類方法」の規定

GIS へ受け渡す地物を CAD で描画するにあたっての「レイヤ分類方法（レイヤへの格納方法）」を規定する。

(4) 「属性」の規定

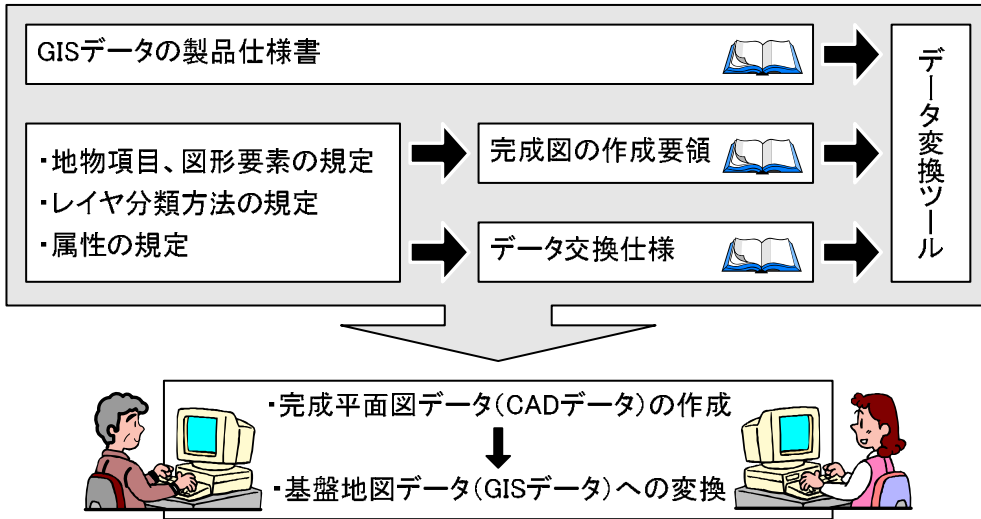
GIS へ受け渡す地物を CAD で作成するにあたって、維持管理での利用用途に応じて各地物に付与する「属性」を規定する。

(5) 「完成図の作成要領」及び「データ交換仕様」の作成

上記（2）～（4）を踏まえ、GIS へ受け渡す地物及び属性情報の CAD（完成図）での作成方法を規定した「完成図の作成要領」及び「データ交換仕様」を作成する。

(6) CAD-GIS 間の「データ変換ツール」の作成

上記（1）及び（5）に基づき、完成図（CAD データ）を GIS データに変換するための「データ変換ツール」を作成する。



4. データ変換にあたっての基本的規定事項

4.1 完成図データの作成について

4.1.1 ファイル形式

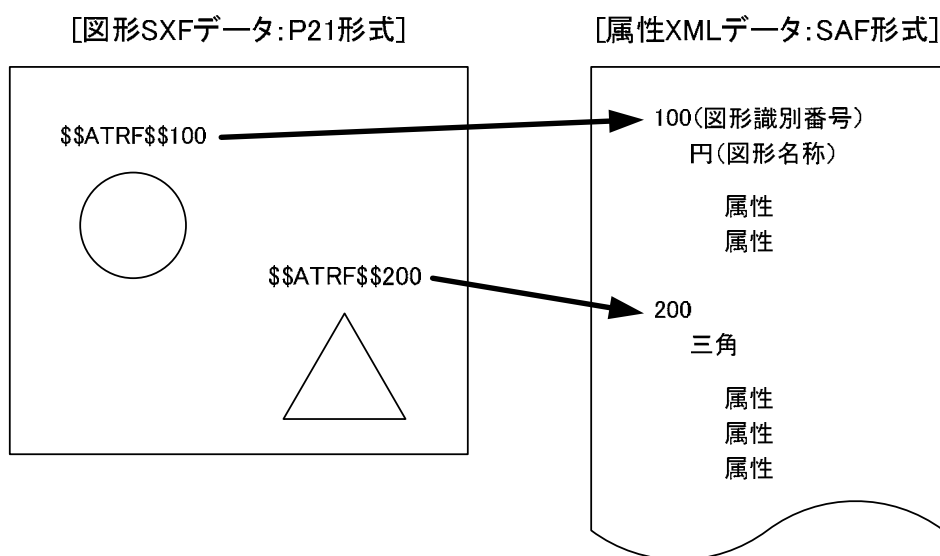
完成図のファイル形式は、図形データに任意の属性データを付与することができる SXF Ver.3.1 の仕様に準拠するものとする。図形データ等を格納するファイルは、P21 形式、属性データを格納するファイルは、SAF 形式とする。

【解説】

完成図のファイル形式は、維持管理段階で GIS に活用することを踏まえ、図形データに任意の属性データを付与することができる SXF Ver.3.1 の仕様に準拠するものとする。

SXF Ver.3.1 の仕様における物理ファイル形式は、図形データ等を格納する P21 形式と図形に対応した属性データを格納する SAF 形式の 2 つにより構成される。

- 図形 SXF データ：P21 形式（拡張子 .p21）
- 属性 XML データ：SAF 形式（拡張子 .saf）



また、ファイル単位は、原則、1 工事 1 ファイルとするが、ファイルサイズが大きすぎることで、作成データの品質が落ちる可能性もあるので、現実的なファイルサイズの上限值を設定する必要がある。

4.1.2 座標系

完成図の座標系は、世界測地系の平面直角座標とする。

【解説】

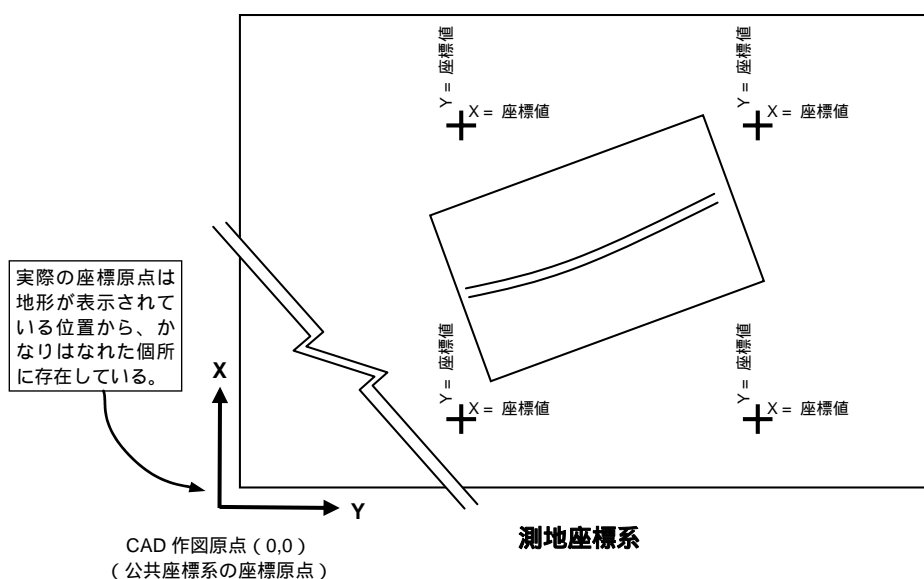
CAD では、用紙座標系での作図を基本とするが、平成 14 年 4 月 1 日の測量法改正以後、すべての測量業務は、世界測地系によるものとされており、完成図の作成に用いる座標系は、世界測地系の平面直角座標系とする。

SXF Ver.3.1 仕様書・同解説では、「用紙座標系」の他に、2 種類の座標系（「数学座標系」と「測地座標系」）を部分図ごとに使用できるものと規定しているが、完成図の作成における座標設定は、「測地座標系」を基本とする。

- 「用紙座標系」：用紙（矩形）の左下を原点（0,0）、単位を mm とする座標系
- 「数学座標系」：直交座標において、東西方向（0 度方向の軸）を X 軸、南北方向（90 度方向の軸）を Y 軸、単位を mm とする座標系
- 「測地座標系」：直交座標において、南北方向（90 度方向の軸）を X 軸、東西方向（0 度方向の軸）を Y 軸、単位を mm とする座標系

また、SXF Ver.3.1 仕様書・同解説では、部分図を用紙上に配置する際には、各部分図の持つ座標系の原点（0,0）を、用紙座標系の任意の位置に指定することで、部分図の用紙上の位置を特定としているので、原点の設定では、部分図の原点（0,0）を地域で定められた平面直角座標系の原点（0,0）と一致させることとする。

さらに、作図は実寸で行い、単位は、SXF Ver.3.1 仕様書・同解説に基づいて mm とする。



4.1.3 作成する図形

GIS データに基づき、完成図で取得対象とする地物項目（図形名称）及び図形要素（図形の種類）（点データ、線データ、面データ）を規定する。対象となる地物項目などは、GIS の利用用途によって異なる。

【解説】

完成図で取得対象とする地物項目は、維持管理段階で利用する GIS アプリケーションで必要となる共有空間データ（基盤地図データ）として「GIS データの製品仕様書」で定める地物とする。なお、維持管理段階で必要となる地物については基本的には取得対象とすることが望ましいが、対象地物数が多くなり過ぎると、データ作成者の負担が大きくなってしまふ。対象とする地物項目の規程にあたっては、データ変換利用による効率化という本来の目的に留意する必要がある。

国土交通省における道路工事の完成平面図の作成では、取得対象とする地物項目は、道路管理者がサービスを実現する上で必要となる共用性の高い情報として「道路基盤データ製品仕様書（案）」が定める 29 地物としており、施工工区の道路区域内におけるすべての地物項目を取得するものとしている。

国土交通省の道路工事の完成平面図における取得対象地物項目

図形名称（地物）	図形要素		
	面データ	線データ	点データ
道路中心線			
距離標			
管理区域界			
道路面	車道部		
	車道交差部		
	踏切道		
	軌道敷		
	島		
	路面電車停留所		
	歩道部		
	自転車駐車場 自動車駐車場		
植栽			
区画線			
停止線			
横断歩道			
横断歩道橋			
地下横断歩道			
建築物			
橋脚			
盛土法面			
切土法面			
斜面对策工			
擁壁			
ボックスカルバート			
シェッド			
シェルター			
橋梁			
トンネル			

「道路工事完成図等作成要領」、「道路基盤データ交換属性セット（案）」(国土交通省)

4.1.4 レイヤ分類

レイヤ分類（レイヤ名、着色）は、CAD 製図基準（案）の規定に準拠することを基本とする。CAD 製図基準（案）の規定では、同一レイヤに複数種類の地物を格納することになるが、データ作成方法のわかりやすさや、目視チェックの容易さ等を重視する場合、地物の種類ごとにレイヤを分類する方法も考えられる。

【解説】

「CAD 製図基準（案）」で規定されているレイヤ分類は、レイヤ単位の色や線種、画面上の表示、紙への出力の設定等、主に図面としての閲覧や出図の作業性を重視したものとされており、同一レイヤに複数種類の地物を格納することになる。

ただし、「複数種類の地物を同一レイヤに格納する方法」は、データ作成の作業量やエラー発生の可能性が少ないが、作業内容がわかりにくく、目視によるチェックがしにくいこともあり、「地物の種類ごとにレイヤを分類する方法」とすることも勘案し、レイヤ分類を決める必要がある。

レイヤ分類方法の比較

		地物の種類ごとにレイヤを分類 (1地物1レイヤ)	複数種類の地物を 同一レイヤに格納
作成方法		<ul style="list-style-type: none"> ・地物ごとにレイヤを分けて境界線を作成 ・面データは、SXF Ver.3.0の機能である既定義ハッチング（Area_control）で作成 ・属性は、ハッチングごとに入力 	<ul style="list-style-type: none"> ・複数地物の境界線を1レイヤ内に作成 ・面データは、SXF Ver.3.0の機能である既定義ハッチング（Area_control）で作成 ・属性は、ハッチングごとに入力
CADによるデータ作成	作成方法の理解	<p>理解が容易</p> <p>1地物ごとに面データを完成させていくため、面を作るイメージがしやすい。</p>	<p>理解しづらい</p> <p>境界線を共通利用しながら面データを作成する作業の内容がわかりづらい。</p>
	作成・編集作業	<p>作業量が多い</p> <p>境界線が重複する箇所、作成・編集作業が二重に必要。</p>	<p>作業量が少ない</p> <p>境界線の重複がないため、作図・編集作業が最小限。</p>
		<p>作業が煩雑</p> <p>異なるレイヤ間で、境界線の端点座標や境界線形状の調整が必要。</p>	<p>作業が単純</p> <p>同一レイヤ内で、境界線の端点座標を一致させるのみ。</p>
		<p>入力量が少ない</p> <p>属性入力は、1項目（設置日）のみ。</p>	<p>入力量が多い</p> <p>属性入力は、2項目（地物名と設置日）。</p>
	目視チェック	<p>チェックしやすい</p> <p>レイヤ切替え表示により、地物ごとの図形データ表示・確認が容易。</p>	<p>チェックしにくい</p> <p>CADソフトに対応機能がない場合、地物ごとの図形データの表示が困難。</p>
品質	<p>修正・手戻り発生の可能性が高い</p> <p>境界線の端点不一致、境界線の形状不一致など、エラー要因が多い。</p>	<p>修正・手戻り発生の可能性が低い</p> <p>境界線の端点不一致にエラー要因が限定される。</p>	

GIS 側の負担	システムへの負担が大きい 不一致境界線のデータを多く含む場合、変換したデータ量が増大し、システムへの負担が増加することで、利便性低下につながる可能性がある。	システムへの負担が小さい 境界線のデータ量が増加する可能性は低く、システムへの負荷小。
特徴	<ul style="list-style-type: none"> ・ 作成者側で作業方法のイメージ・理解がしやすい方法 ・ エラー発生の可能性が高いため、エラー低減、品質向上に向けたフォローが必要 	<ul style="list-style-type: none"> ・ 編集作業量、エラー発生の可能性が少ない方法 ・ 作業の不慣れ、わかりにくさの解消に向けたフォローが必要

例示：道路工事完成図における試行運用の際の評価

地物の種類ごとにレイヤを分類した場合は、レイヤ数が多くなることから、独自のレイヤ命名則を規定する必要がある。また、データを目視等で確認できるように、着色（表示色）も規定する必要がある。

国土交通省における道路工事の完成平面図の作成では、「地物の種類ごとにレイヤを分類する方法」と「複数種類の地物を同一レイヤに格納する方法」を比較した結果、データ作成の作業量や作業の煩雑さでは劣るものの、実施する作業内容がイメージしやすく、総合的に判断して作業が行いやすい方法として、「地物の種類ごとにレイヤを分類する方法」を採用した。

なお、地物ごとのレイヤ分類とすることにより、レイヤ数が多くなることを考慮し、レイヤ命名の規則については3階層目までのレイヤ名称をCAD製図基準（案）に準拠しながら、4階層目に地物名称を表す英語表記を採用するものとしている。



国土交通省の道路工事の完成平面図におけるレイヤ名の例

つまり、各地物データの作成とチェックの作業性に重点を置くものとし、地物と格納するレイヤの対応関係を明確にするため、一地物ごとにレイヤ分類を行うとともに、土木学会の「土木CAD製図基準」を参考に4階層のレイヤ命名則を採用し、格納する地物を連想可能なレイヤ名としている。CAD製図基準（案）では4階層のレイヤ命名則を採用していないため、国土交通省（「道路工事完成図等作成要領」）の独自規定である。

国土交通省の道路工事の完成平面図におけるレイヤ分類

図形名称（地物）		レイヤ名	着色
道路中心線		C-BMK	黄
距離標		C-BMK-BMKZ-KMPOST	緑
管理区域界		C-BMK-BMKZ-BOUNDARY	橙
道路面	車道部	C-STR-STRZ-ROADWAY	暗灰
	車道交差部	C-STR-STRZ-CROSSING	暗灰
	踏切道	C-STR-STRZ-RAILROADCROSS	赤
	軌道敷	C-STR-STRZ-TRAMAREA	赤
	島	C-STR-STRZ-ISLAND	黄緑
	路面電車停留所	C-STR-STRZ-TRAMSTOP	赤
	歩道部	C-STR-STRZ-SIDEWALK	桃
	自転車駐車場	C-STR-STRZ-BICYCLEPARK	暗灰
	自動車駐車場	C-STR-STRZ-CARPARK	暗灰
植栽		C-STR-STRZ-PLANT	黄緑
区画線		C-STR-STRZ-LINE	白
停止線		C-STR-STRZ-STOPLINE	白
横断歩道		C-STR-STRZ-CROSSWALK	白
横断歩道橋		C-STR-STRZ-PEDESTRIANBRIDGE	赤
地下横断歩道		C-STR-STRZ-UNDERPASSWALK	赤
建築物		C-STR-STRZ-BUILDING	茶
橋脚		C-STR-STRZ-PIER	赤
盛土法面		C-STR-STRZ-BANK	緑
切土法面		C-STR-STRZ-CUT	橙
斜面对策工		C-STR-STRZ-SLOPE	赤
擁壁		C-STR-STRZ-WALL	赤
ボックスカルバート		C-STR-STRZ-BOX	赤
シェッド		C-STR-STRZ-SHED	赤
シェルター		C-STR-STRZ-SHELTER	赤
橋梁		C-STR-STRZ-BRIDGE	赤
トンネル		C-STR-STRZ-TUNNEL	紫

「道路工事完成図等作成要領」、「道路基盤データ交換属性セット（案）」（国土交通省）

4.1.5 属性入力

GIS データに基づき、作成する図形データ（各地物）に入力する属性項目を規定する。対象とする属性項目は、GIS の利用用途によって異なる。

【解説】

完作成する図形データ（各地物）に入力する属性項目は、「GIS データの製品仕様書」で定める属性へ変換するデータとする。なお、維持管理段階で必要となる属性については基本的には入力の対象とすることが望ましいが、入力する属性項目数が多くなり過ぎると、データ作成者の負担が大きくなってしまう。対象とする属性項目の規程にあたっては、データ変換利用による効率化という本来の目的に留意する必要がある。

同一レイヤに複数種類の地物を格納する場合は、地物名称が属性項目にないと地物種類を特定できないが、国土交通省における道路工事の完成平面図の作成では、地物の種類ごとにレイヤを分類（1地物1レイヤ）することで、レイヤ名称から地物種類を特定可能としている。

国土交通省の道路工事の完成平面図における入力属性項目

地物名	入力する属性項目
距離標	設置日 路線番号 現旧区分 上下区分 接頭文字 距離程 種別 経度 緯度 高さ
距離標以外の地物	設置日

「道路工事完成図等作成要領」、「道路基盤データ交換属性セット（案）」（国土交通省）

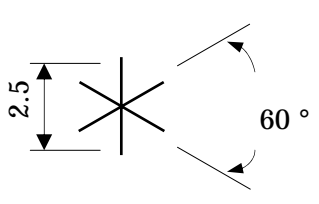
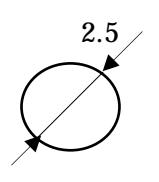

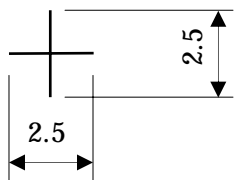
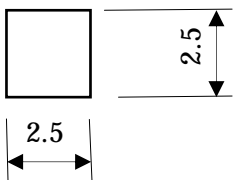
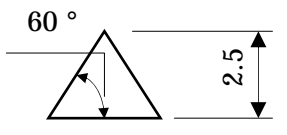
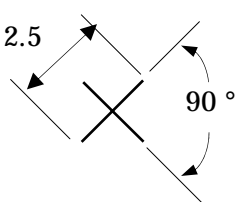
4.1.6 点データの作成

点データは、SXF Ver.3.1 仕様書・同解説における点マーカを用いるものとする。

【解説】

SXF Ver.3.1 仕様書・同解説では、点マーカのマーカコードとして、以下のようなものが示されている。点の表示方法は、視認性が確保できればよく、用いるマーカコードは、任意としても、指定（表示の大きさを含め）してもかまわない。

SXF Ver.3.1 における点マーカのマーカコード

<p>1: asterisk</p> 	<p>2: circle</p> 	<p>3: dot</p> 
<p>4: plus</p> 	<p>5: square</p> 	<p>6: triangle</p> 
<p>7: X</p> 		

点の色については、GIS データに含まれないデータであり、変換しないことから、任意としてもかまわないが、目視によるチェックができるように、レイヤ分類で規定した着色に従うこととする。

点データで作成する地物の図形は、1 地物に対して 1 つの図形とし、位置座標 (x, y) 1 点のみを持つデータとして作成する。

国土交通省における道路工事の完成平面図の作成では、点データのマーカコードは、「5: square」とし、表示の大きさは、2 mm（印刷時）を標準としている。

4.1.7 線データの作成

線データの作成で利用可能な図形要素（直線（線分、折線）、円弧、楕円弧、スプライン曲線）を規定する。

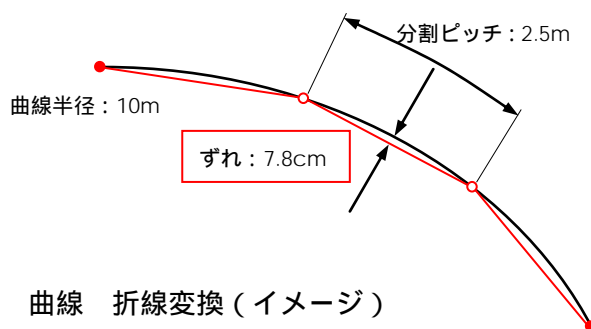
【解説】

CADの曲線を、GISでは曲線として認識できないことから、CAD（完成図データの作成）又はGIS（データ変換のツール）で連続した線分（折線）に変換する必要がある。

このことを踏まえて線データの作成で利用可能な図形要素を規定する。

線データは、すべて直線（折線、線分）で作成することとするか、曲線の一部（例えば、円弧）も可とするかについては、CAD図面の再利用性、作図時の一般性などを勘案し、決める。

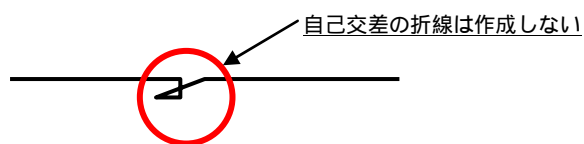
円弧、楕円弧、スプライン曲線といった図面要素を用いないこととする場合は、CAD側で曲線を折線に変換することとなるので、曲線の折線変換時の相対誤差を規定する必要がある。



国土交通省における道路工事の完成平面図の作成では、利用可能な図形要素は、直線と円弧としている。発注図で既に作成されているデータを極力そのまま活用し、必要最低限の編集作業で完成平面図を作成できるように、線データ及び面データの境界線作成時に、既存作図線を利用できるようにするため、発注図の基となる道路設計の図面作成で多く利用されている直線と円弧を利用可能としている。なお、その他の曲線（楕円弧、スプライン曲線）については、道路設計で一般に用いられることが少ないため、利用不可としている。

CAD 製図基準（案）では、線種（実線、破線、一点鎖線、二点鎖線の 4 種類）、太さ（細線、太線、極太線の 3 種類）、色が規定されているが、GIS データに含まれないデータであり、変換しないことから、任意として構わないが、線の色については、目視によるチェックができるように、レイヤ分類で規定した着色に従うこととする。

線データで作成する地物の図形は、1 地物に対して 1 つの図形とする。GIS データへの変換を適性に行うため、分割・途切れやねじれ構造（自己交差）のない線データとする。



4.1.8 面データの作成

面データの作成にあたっては、境界線図形で幾何位相を表現する方法と、一致する形状で幾何位相を表現する方法があるが、作図作業を行いやすいことから、一致する形状で幾何位相を表現する方法を用いることを基本とする。

【解説】

面データの作成については、領域の外周を表す境界線データと領域内の既定義ハッチング (Area_control) を組み合わせて作成する方法の方が、GIS データへの変換がしやすいが、作図作業が難しく、エラーの発生の可能性が高い。

面データの作成にあたっては、作図作業を行いやすい、一致する形状で幾何位相を表現する方法を用いることを基本とする。

位相構造 (幾何位相) の比較

	境界線図形で幾何位相を表現する方法	一致する形状で幾何位相を表現する方法
作成方法	領域の外周を表す境界線データと領域内の既定義ハッチング (Area_control) を組み合わせて作成する	既定義ハッチング (Area_control) のみを用いて作成する。
イメージ		
メリット	<ul style="list-style-type: none"> 幾何位相を構築する境界線が、明示的に作図されているので、データ変換しやすい。 隣り合う面データの境界線が 1 本であれば、位相構築の品質は高い。 	<ul style="list-style-type: none"> ハッチング作成のための境界線作図が不要となり、端点位置の分割、位置補正などが少なくてすむ。 既存の作図線が利用しやすい。
デメリット	<ul style="list-style-type: none"> 1 地物 1 レイヤとした場合は、同一形状の境界線を複数作成する必要があり、作業効率が悪い。 	<ul style="list-style-type: none"> 複合曲線の境界形状一致が前提であるので、一致しない場合は、位相構造としてみなすことができない。

国土交通省における道路工事の完成平面図の作成においても、位相構造 (幾何位相) とする面データの作成方法として、「一致する形状で幾何位相を表現する方法 (ハッチングの複合曲線形状の形状一致により表す方法)」を採用している。

参考： 国土交通省の道路工事完成図等作成要領における面データ作成のポイント

上記の、一致する形状で幾何位相を表現する方法で面データ作成するにあたり、ハッチングを用いた作図を行うこととなりますが、既定義ハッチング (Area_control) は無色透明であり目視ができないため、この既定義ハッチング (Area_control) を用いたデータ作成について、施工業者が対応することは実質的には困難でした。そこで、道路工事完成図等作成要領では、作図の際には色つきの (通常) のハッチングを用い、SXF 出力する際に CAD の機能を用いて Area_control へ変換することで、データ作成者の作業負担を軽減すると共に、データ作成ミスの低減を図っています。

4.2 GIS データへの変換について

4.2.1 データ変換ツール

完成図（CAD データ）を GIS データへ変換するため、CAD-GIS 間の「データ変換ツール」を作成する。

【解説】

完成図（CAD データ）の GIS データへの変換にあたり、CAD-GIS 間の「データ変換ツール」の作成が必要である。

このツールの作成は、「3. CAD-GIS 連携の手順」に従って作成する「完成図の作成要領」、「データ変換仕様」及び「GIS データ製品仕様書」に基づくものとする。

なお、国土交通省国土技術政策総合研究所の道路工事完成図等作成支援サイトのダウンロードのページ（http://www.nilim-cdrw.jp/dl_other.html）より「CAD-GIS コンバータ【道路版】」の「プログラム本体」と「設計書およびプログラム仕様書」が入手可能であり、変換ツール作成の参考にすることができる。

4.2.2 データ変換ツールの基本的な機能

データ変換ツールに必要となる基本的な機能は以下のとおりである。

- (1) 図形データの読み込み機能
- (2) 空間属性への変換機能
- (3) 属性データの読み込み機能
- (4) 主題・時間属性の変換機能
- (5) GIS データの出力機能

【解説】

それぞれの機能については、以下のとおりである。

(1) 図形データの読み込み機能

図形 SXF データ（P21 形式）を読み込む。

【解説】

図形 SXF データ（P21 形式）を読み込む機能については、「SXF Ver.3.1 対応レベル 2 共通ライブラリ」の利用が可能である。

このライブラリは、国土交通省の CALS/EC 電子納品に関する要領・基準サイト（<http://www.cals-ed.jp>）より入手可能である。

(2) 空間属性への変換機能

図形 SXF データ (P21 形式) より得られた作図情報を空間属性に変換する。曲線の折線への変換、緯度経度座標への座標変換、図形識別番号 (ID) の変換、幾何位相の構築・変換を行う。

【解説】

1) 曲線の折線への変換

完成図の線データの作成で、曲線 (円弧、楕円弧、スプライン) を可としている場合は、規定した相対誤差以内の折線に変換する。

2) 緯度経度座標への座標変換

完成図データ (CAD データ) で採用している座標値が平面直角座標であるので、緯度経度に変換する。

3) 図形識別番号 (ID) の変換

完成図データ (CAD データ) では、図形 SXF データ (P21 形式) と属性 XML データ (SAF 形式) のリンクキーとして、「図形識別番号」を用いており、それを図形の ID として利用している。GIS データ (JPGIS 対応 XML) では、空間属性 (図形) と地物が別の ID を利用していること、空間属性の参照関係が P21 形式で記述される図形構造と異なることから、SXF の情報として保有していない ID を新規付与する。

4) 幾何位相の構築・変換

完成図データでは、幾何位相を示す明示的な情報を保有していないため、GIS データに変換する際に、幾何位相を構築する。

レイヤ、線種・線色・線幅などの GIS データに含まれていないデータは、変換しない。

(3) 属性データの読み込み機能

属性 XML データ (SAF 形式) を読み込む。

【解説】

属性 XML データ (SAF 形式) を読み込む。

(4) 主題・時間属性の変換機能

対応するタグ同士で変換する。

【解説】

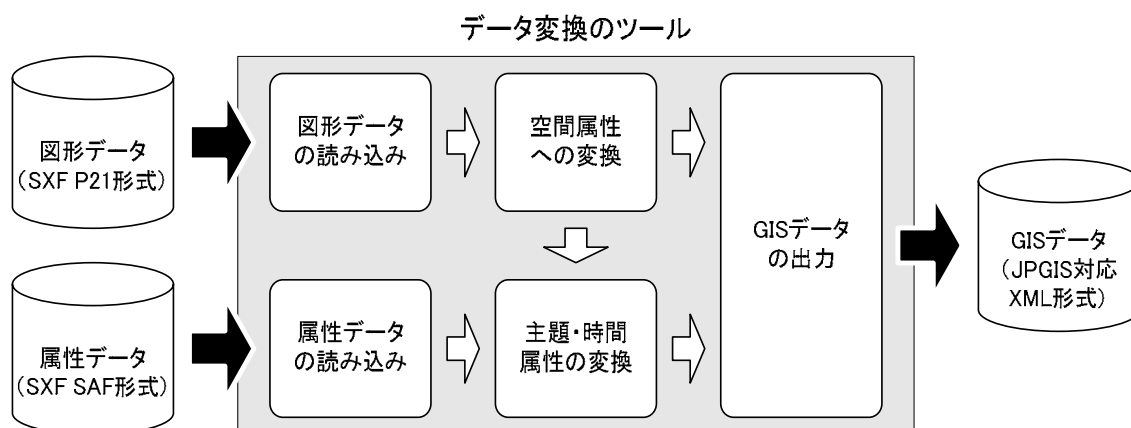
主題・時間属性の記述は、同じ XML による記述であることから、対応するタグ同士で変換する。

(5) GIS データの出力機能

空間属性、主題、時間属性を JPGIS 対応 XML 形式として書き出す。

【解説】

空間属性、主題、時間属性を GIS データ (JPGIS 対応 XML 形式) として書き出す。



国土技術政策総合研究所資料

TECHNICAL NOTE of NILIM

No.507

January 2009

編集・発行 ©国土技術政策総合研究所

本資料の転載・複写の問い合わせは

〒305-0804 茨城県つくば市旭1番地

企画部研究評価・推進課 TEL 029-864-2675