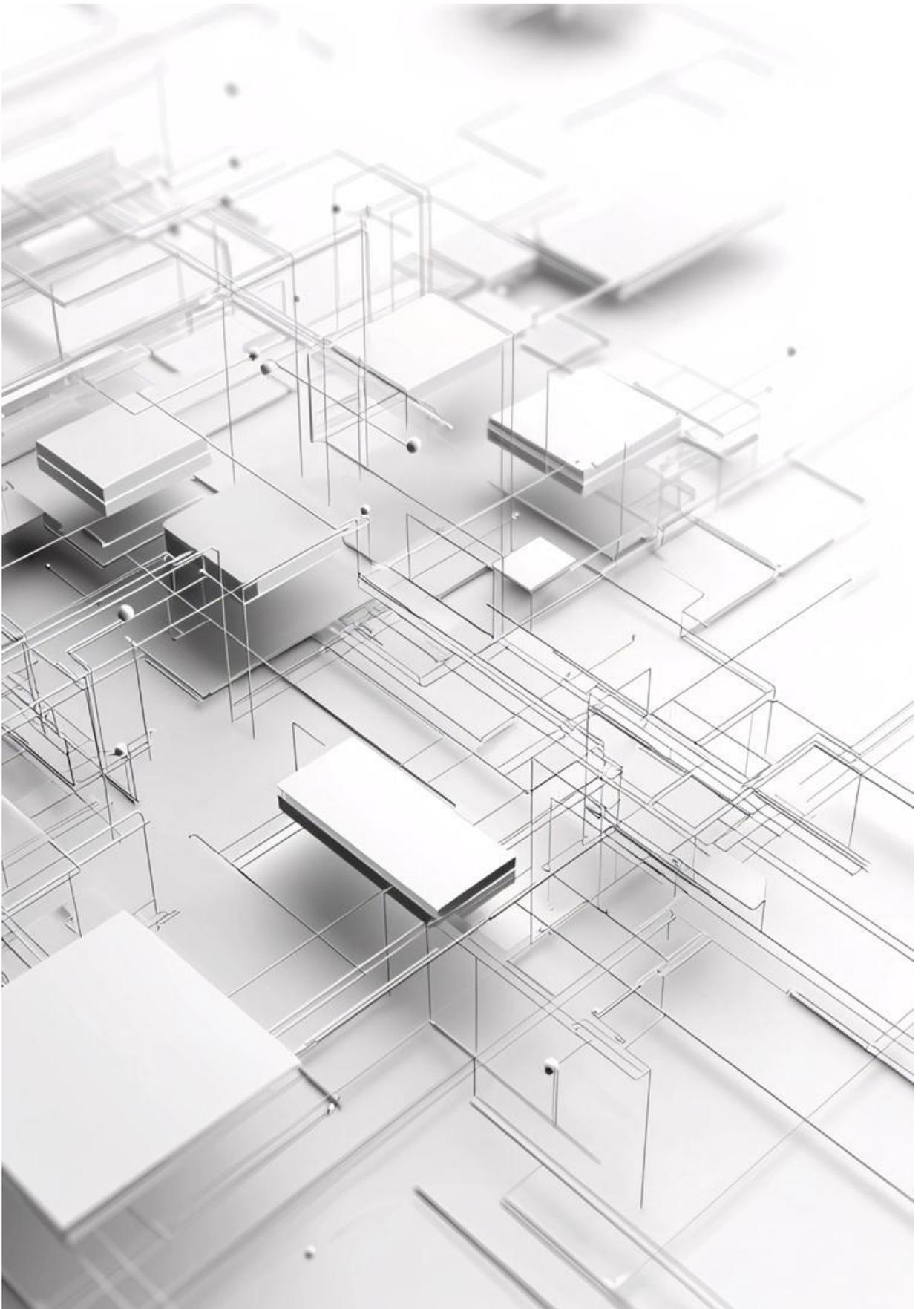




PLATEAU  
by MLIT

PLATEAU Technical Report  
3D都市モデル活用のための技術資料



モデル駆動型アーキテクチャによる3D都市モデル  
標準ドキュメントの管理システム 技術検証レポート

Technical Report on MDA (Model-Driven Architecture) for PLATEAU Documentation

series  
No. **114**

# 目次

---

---

1. プロジェクトの概要 .....	- 3 -
1-1. 現状と課題 .....	- 3 -
1-1-1. ドキュメント生成の既存フロー .....	- 3 -
1-1-2. 既存フローへの課題認識 .....	- 4 -
1-2. 課題解決のアプローチとその創出価値 .....	- 4 -
1-3. 本プロジェクトの対応範囲 .....	- 4 -
2. 可用性向上のための MDA 導入の概要 .....	- 5 -
2-1. モデル駆動型アーキテクチャ導入 .....	- 5 -
2-2. 利用するソフトウェアと構築内容 .....	- 6 -
2-2-1. Enterprise Architect .....	- 6 -
2-2-2. ShapeChange .....	- 6 -
2-2-3. Metanorma .....	- 7 -
2-3. ドキュメント生成の MDA 導入フロー .....	- 8 -
2-4. 検証ポイント .....	- 9 -
2-5. スケジュール .....	- 10 -
3. 開発スコープ .....	- 11 -
3-1. 概要 .....	- 12 -
3-2. 開発内容 .....	- 13 -
3-2-1. 概念モデル作成における開発 .....	- 13 -
3-2-2. 符号化仕様作成における開発 .....	- 14 -
3-2-3. ドキュメントのスタイル作成における開発 .....	- 15 -
3-2-4. LutaML のコマンドとレンダリングスタイルの変更 .....	- 18 -
3-2-5. ドキュメントのコンテンツ作成における開発 .....	- 19 -
3-3. 開発対象の GitHub 管理 .....	- 21 -
4. 実証システム .....	- 22 -
4-1. アーキテクチャ .....	- 22 -
4-2. 各ソフトウェアの機能と入出力されるファイルフロー .....	- 23 -
4-2-1. 利用ソフトウェアと環境要件 .....	- 23 -
4-2-2. 機能と入出力されるファイル .....	- 24 -
4-2-3. データフロー .....	- 25 -
4-3. 各機能の利用手順 .....	- 26 -
4-3-1. EA のプロジェクトファイル変換 .....	- 26 -
4-3-2. EA のモデル・ダイアグラム画像出力 .....	- 26 -
4-3-3. ShapeChange の XSD 出力 .....	- 27 -
4-3-4. ExcelVBA による XSD 整形 .....	- 27 -

4-3-5. Metanorma によるドキュメント生成 .....	- 28 -
5. 品質.....	- 29 -
5-1. 機能要件の品質担保 .....	- 30 -
5-2. 非機能要件の品質担保.....	- 32 -
6. 実証技術の機能要件の品質検証 .....	- 33 -
6-1. 概念モデルと符号化仕様の作成検証 .....	- 33 -
6-1-1. 検証方法 .....	- 33 -
6-1-2. 検証結果 (XSD 変換) .....	- 34 -
6-1-3. 検証結果 (XSD 整形) .....	- 38 -
6-2. 概念モデルとドキュメント生成の検証 .....	- 44 -
6-2-1. 検証方法 .....	- 44 -
6-2-2. 検証結果 (EA と LutaML) .....	- 44 -
6-2-3. 検証結果 (スタイル) .....	- 48 -
6-2-4. 検証結果 (コンテンツ) .....	- 51 -
7. 実証技術の非機能要件の品質検証 .....	- 52 -
7-1. 概念モデルと符号化仕様の作成検証 .....	- 52 -
7-1-1. 検証方法 .....	- 52 -
7-1-2. 検証結果 (XSD 整形) .....	- 52 -
7-2. 概念モデルとドキュメント生成の検証 .....	- 53 -
7-2-1. 検証方法 .....	- 53 -
7-2-2. 検証結果 (ドキュメント生成) .....	- 54 -
8. 成果と課題 .....	- 55 -
9. 参考資料.....	- 56 -
10. 用語 .....	- 57 -

# 1. プロジェクトの概要

## 1-1. 現状と課題

3D 都市モデル標準製品仕様書（以下、「標準製品仕様書」と略す）の対象地物拡充に伴い製品仕様が複雑化しており、さらにそのドキュメント化の過程で複数のファイルを手作業で更新・統合しているため、人為的エラーが生じやすい状況である。

このためドキュメント化の工程を見直し、標準製品仕様書をはじめとした標準ドキュメントの可用性を向上させることを検討する。

### 1-1-1. ドキュメント生成の既存フロー

標準製品仕様書のドキュメント生成の既存フローは下図のとおりである。

XML スキーマは XSD ファイル（以下、「XSD」と略す）として作成され、定義文書は標準製品仕様書において応用スキーマ文書として表現される。

また 3D 都市モデル標準作業手順書（以下、「作業手順書」と略す）は、標準製品仕様書と一部内容が重複するにもかかわらず、以下のフローとは完全に独立・並行して、Word と PDF 及び HTML を作成している。

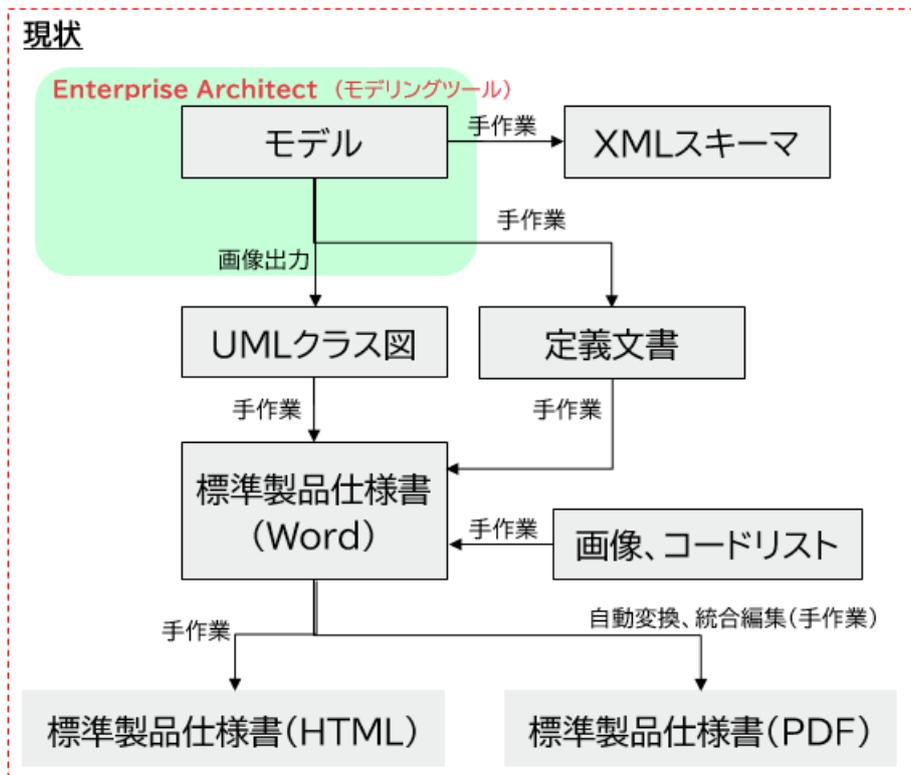


図 1-1 標準製品仕様書のドキュメント生成の既存フロー

## 1-1-2. 既存フローへの課題認識

既存フローの課題と想定する対策は、以下のとおりである。

表 1-1 既存フローの課題と想定する対策

課題	対策
記載内容が量・質ともに膨大で複雑である	工程の自動化を図る
成果物（XSD、標準製品仕様書、作業手順書）を別々に作成するため不整合が発生しやすい	ソース情報を可能な限り一元化する

## 1-2. 課題解決のアプローチとその創出価値

課題対策として、ドキュメント生成フローへのモデル駆動型アーキテクチャ（Model Driven Architecture：以下、「MDA」と略す）の導入を図る。

MDA は、ソフトウェア設計・開発・実装手法の一種であり、実装方法に依存しないモデルから、実装となるプログラムを生成する仕組みである。

MDA をドキュメント化のフローに導入することによって、モデリングツール等のソース情報から、具体的な実装となる PDF や HTML といった各種形式のドキュメントを、自動生成する仕組みを構築する。

これによって可用性を向上させ、ドキュメント生成フローにおける整合性と効率性を確保し、「コストの削減」、「人為エラーの軽減と手戻り防止」、「品質と保守性の向上」等の価値を実現することを目指す。

## 1-3. 本プロジェクトの対応範囲

標準製品仕様について、可用性向上のプロジェクトと並行し、利便性向上のプロジェクトと有用性向上のプロジェクトを進める。

各プロジェクトの内容は下記のとおりとし、可用性向上では標準製品仕様の改訂や前提知識には踏み込まない。

表 1-2 標準製品仕様に関するプロジェクト

プロジェクト	内容
可用性向上	標準製品仕様を表現・定義するドキュメント生成に MDA を導入する。
利便性向上	標準製品仕様とその前提知識の理解を助けるサブドキュメント群を構築する。
有用性向上	標準製品仕様の調査・検討によって、標準製品仕様書の改訂を行う。

## 2. 可用性向上のための MDA 導入の概要

### 2-1. モデル駆動型アーキテクチャ導入

MDA を導入した標準ドキュメントの生成において、対応すべきポイントは下記のとおりである。

表 2-1 MDA 導入の対応すべきポイント

ポイント	内容
①	概念モデルとなる UML クラス図とその定義文書に相違がないこと。
②	概念モデルと符号化仕様（ドキュメント生成フローでは XML スキーマ）が一致すること。
③	概念モデルや符号化仕様及びその他の必要な情報を一体的に管理し、ドキュメント化すること。

上記の各ポイントを満たしたドキュメント生成の先行事例として、Open Geospatial Consortium（以下、「OGC」と略す）の標準ドキュメント生成がある。この先行事例を参考とし（「9.参考資料」を参照）、各ポイントの達成を、既存フローでクラス図画像の出力に利用している Enterprise Architect のさらなる活用、及びフリーのソフトウェアである ShapeChange と Metanorma を導入することで実現する。

表 2-2 MDA 導入の対応と利用するソフトウェア

ポイント	対応	利用するソフトウェア
①	UML クラス図をアップデートしてクラス等の定義を入力するとともに、入力された情報を使って定義文書を作成する。	Enterprise Architect
②	Enterprise Architect から、XSD を自動生成する。	ShapeChange
③	Enterprise Architect から出力する UML クラス図や定義文章と、標準文書に必要なその他の情報（画像やコードリスト等）を統合し、標準文書として整形し、PDF/HTML など必要な文書形式に出力する。	Metanorma

## 2-2. 利用するソフトウェアと構築内容

表 2-2 で記載したソフトウェアの概要と、ドキュメント生成フローで利用するために必要な構築内容は下記のとおりである。

### 2-2-1. Enterprise Architect

Enterprise Architect (以下、「EA」と略す) は、オーストラリアの SparxSystems 社が開発・販売している UML モデリングツールである。EA は OGC でも採用されており、CityGML 2.0 やこれが引用する他の標準の UML クラス図が EA を使って作成されていること、また EA から UML クラス図を、メタデータ情報の交換標準である XML Metadata Interchange 形式 (以下、「XMI」と略す) に出力する機能を有することから、3D 都市モデルの MDA 導入においても採用することとした。

既存フローでは EA を UML クラス図の画像出力で用いていたため、標準製品仕様のためのプロジェクトファイルでは、一部定義が最新化されていない状態である。このため、定義の最新化、及び有用性向上のプロジェクトによる標準製品仕様の改訂を反映する必要がある (反映内容については本書の対応範囲外とする)。あわせて、ShapeChange や Metanorma で利用できるよう、設定調整や定義文の追記を行う。

### 2-2-2. ShapeChange

ShapeChange は、XML 形式で記載される Config 設定に沿って、概念モデルを符号化仕様に変換するフリーツールである。Java で開発されており、実行ファイルは JRE11 の環境で動く JAR ファイルである (本開発時点のバージョンは 3.1.0 である)。

ShapeChange も EA 同様、OGC においても採用されており、CityGML3.0 の XSD も ShapeChange を用いて同様の手順で生成出力されている。

なお ShapeChange で出力した XSD は、現行の記載方法が合致しない点が発生するため、整形プログラムを別途開発することで対応する。

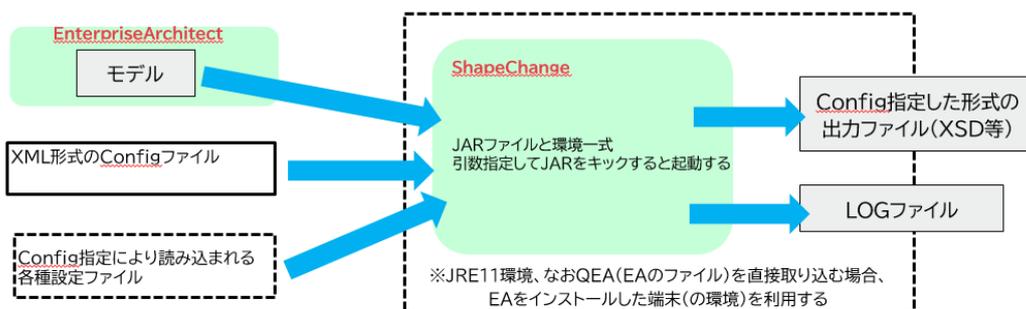


図 2-1 ShapeChange 出力のイメージ

### 2-2-3. Metanorma

Metanorma は、標準文書を作成・発行するため、主として Ruby で開発されたオープンソース・フレームワークであり、香港に拠点を置く Ribose 社によってメンテナンスされている（「9.参考資料」を参照）。Metanorma AsciiDoc 形式（軽量マークアップ言語である AsciiDoc に一部拡張と制限を加えた言語、以下、「Metanorma ADOC」と略す）で記述されたコンテンツから、指定されたスタイル（テンプレート）で出力する方式であり、OGC でも今後は標準文章を Metanorma で作成する方針となっている。

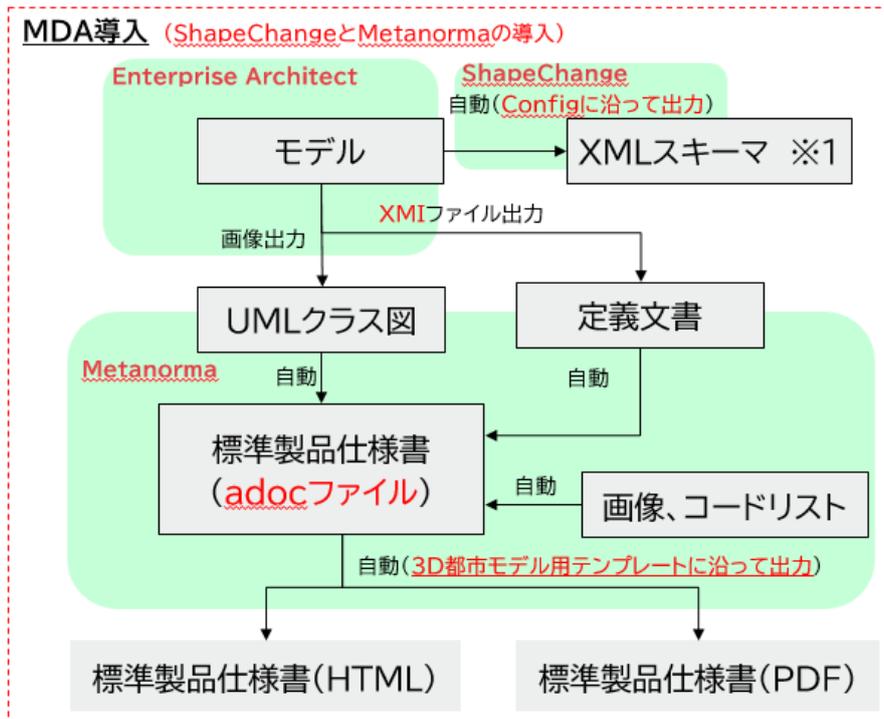
コンテンツである adoc ファイルは、include のコマンドを用いて別の adoc ファイルに組み込んだ形で出力することが可能である。この機能を利用し、章節単位等の内容でファイル分けして、ドキュメントを構造的に管理することが可能である。

また LutaML という拡張機能のコマンドを利用することによって XMI を取り込み、標準製品仕様書の応用スキーマ文書の自動レンダリングを可能とする（コードリストの XML ファイルも同様に LutaML で取り込対応とする）。ただし LutaML の仕様に制約されるため、現行とスタイルが異なる点が発生する。

スタイルについては、既に Ribose 社において開発済みであった ISO/JIS 規格（一部 OGC 規格）のスタイルをベースとし、現行のスタイルと異なる点を改修する。

## 2-3. ドキュメント生成の MDA 導入フロー

標準製品仕様書のドキュメント生成の MDA 導入フローを下図のとおり定義する。  
 有用性向上の成果を反映した EA のモデル情報を用いて、符号化仕様である XML スキーマ（製品に付属する urf と uro の XSD を対象とする）と、標準ドキュメントを生成するフローとなる。  
 また作業手順書も同様に adoc ファイルで構造化することによって、重複するソースの共通利用を可能とする。



※1: 必要に応じ整形プログラムで形式を整える

図 2-2 標準製品仕様書のドキュメント生成の MDA 導入フロー

なお、標準製品仕様書を含む PLATEAU 技術文書の表紙には共通するスタイルが適用されているが、3D 都市モデルのテンプレート適用では同様の表紙を生成できなかったため、PDF 出力に際しては表紙のみ手動で差し替えることとした。(※)

※現状は表紙全体が画像であるが、スタイルの表紙はイラスト部分のみが画像である。現行の表紙画像の文字部分は有償フォントであるため、Metanorma のスタイルに設定できない。

また HTML 出力については、UI の可用性を高めるビューラッパーを被せて表示可能とする。

ともに MDA 導入とは直接関係がないため、上記のフローと本書の対応範囲外とする。

## 2-4. 検証ポイント

本システムのフローについて、検証ポイントを下記のとおり整理する。

図 2-3 MDA 導入フローの検証ポイント

フロー	検証ポイント
概念モデルと 符号化仕様	<ul style="list-style-type: none"> <li>● EA や ShapeChange の Config に反映した修正が、XSD にも反映されるか</li> <li>● 現行 XSD と同内容かつ可能な限り同形式の XSD を出力できるか</li> <li>● ドキュメント出力のオペレーションは容易に実行可能か</li> <li>● ドキュメント出力の速度は極端に遅くないか</li> </ul>
概念モデルと ドキュメント の生成	<ul style="list-style-type: none"> <li>● 開発したスタイルの定義が反映されているか</li> <li>● 応用スキーマ文書とコードリストが現行ドキュメントと可能な限り同内容かつ同スタイルで作成されるか</li> <li>● 現行ドキュメントと同内容かつ可能な限り同スタイルで出力できるか</li> <li>● EA に反映した修正が、ドキュメントにも反映されるか</li> <li>● ドキュメント生成のオペレーションは容易に実行可能か</li> <li>● ドキュメント出力の速度は極端に遅くないか</li> </ul>

## 2-5. スケジュール

本プロジェクトのスケジュールを下記に示す。

図 2-4 スケジュール

実施事項	2024 年										2025 年		
	4 月	5 月	6 月	7 月	8 月	9 月	10 月	11 月	12 月	1 月	2 月	3 月	
1. MDA 導入についての技術調査	←→												
2. 標準製品仕様 4.0 に基づく調査			←→										
3. 標準製品仕様 4.1 に基づく検証						←→							
4. 標準製品仕様 5.0 に基づく出力									←→				
5. レビューとフィードバック反映									←→				
6. 成果取りまとめ										←→			

### 3. 開発スコープ

---

可用性向上において開発対象とするのは、標準仕様ドキュメントの生成に関する機能とする。具体的には XML スキーマ、標準製品仕様書、及び作業手順書の生成に関する機能とする。

したがって objectlist 等の製品に付属するドキュメントのフォーマットファイル作成については対象外とするが、拡張製品仕様書テンプレートについては、標準製品仕様書の Word ファイルに追記して作成するだけなので、Metanorma で出力した標準製品仕様書の PDF から、Adobe Acrobat Pro 等で Word に出力し、原本を作成する。

### 3-1. 概要

ドキュメント生成フローにおける MDA の機能を実現するため、下記の開発を行う。

なお MDA 導入後のドキュメント生成フローを一つの実証システムとして考え、準備作業や一部オペレーションも機能開発に含めるものとする。

表 3-1 機能の開発概要

機能	開発概要
概念モデル作成	<ul style="list-style-type: none"> <li>● ShapeChange や Metanorma で利用できるよう、EA の設定を調整する</li> <li>● 応用スキーマ文書に表示する説明文を実装する (※1)</li> </ul>
符号化仕様作成	<ul style="list-style-type: none"> <li>● EA から XSD を出力するための Config ファイルを開発する</li> <li>● 出力した XSD では、現行の記載ルールと合致しない点が発生するため、対応する整形プログラムを開発する</li> </ul>
ドキュメントのスタイル作成	<ul style="list-style-type: none"> <li>● 現行とのスタイルの相違を踏まえ、ISO/JIS 規格（一部 OGC 規格）を適用した、3D 都市モデル用のテンプレートを開発し (※2)、標準製品仕様書と作業手順書で共用利用可能とする (※3)</li> </ul>
LutaML 新機能開発	<ul style="list-style-type: none"> <li>● 応用スキーマ文書とコードリスト作成のため、3D 都市モデル用に LutaML の新コマンドを開発する (※4)</li> </ul>
ドキュメントのコンテンツ作成	<ul style="list-style-type: none"> <li>● 現行ドキュメントを adoc ファイル一式に構造化して落とし込む (※5)、また標準製品仕様書と作業手順書の同一内容箇所を、adoc で共有利用可能とする</li> <li>● adoc ファイル一式に LutaML を実装する</li> <li>● 拡張製品仕様書テンプレート作成のため、標準製品仕様書の PDF から Adobe Acrobat Pro 等で Word 出力する</li> </ul>

※1：この作業に際して、既存の記載誤りや不整合を修正し、改訂の成果を反映する作業が発生する。有用性向上の範囲であるため、本書の対応範囲外とする。

※2：HTML 出力において、UI の可用性を高めるビューラッパーを開発する。MDA 導入とは直接関係がないため、本書の対応範囲外とする。なお使用フォントは「ビジュアルアイデンティティ (VI) マニュアル第 2.0 版」に準ずる。

※3：PDF 出力に際して、手動で既存の表紙と切り替える。MDA 導入とは関係がないため、本書の対応範囲外とする。

※4：列挙型は adoc ファイルに直書きする。

※5：この作業に際して、現行ドキュメントの不整合を整理し修正する。有用性向上の範囲であるため、本書の対応範囲外とする。

## 3-2. 開発内容

### 3-2-1. 概念モデル作成における開発

EA のプロジェクトファイルは下記のパッケージを有する。

- CityGML2.0 のパッケージ（配下に各地物のサブパッケージを持つ）
- UR3.1 のパッケージ（配下に urf、uro 及び urg、urt のサブパッケージを持つ）
- UML クラス図の管理出力のためのビューパッケージ

さらにプロジェクトファイル内での整合性を保つため、XS と GML のパッケージも有する。

ShapeChange や Metanorma で利用できるよう、各パッケージに対し、下表とおりの変更を加える。

表 3-1 EA プロジェクトファイルの設定変更箇所

変更が必要な箇所	変更内容
画像出力で不要であったパッケージ設定のプロパティは、大半が空白である	i-UR3.1 配下のサブパッケージに対し、ShapeChange の取り込みで最低限必要な ApplicationSchema（定義：GML）の設定を、以下のとおり行う。 <ul style="list-style-type: none"> <li>● xsdDocument に XSD の名称を設定する</li> <li>● targetnamespace にパス設定する</li> <li>● xmlns に urf 又は uro を設定する</li> <li>● version を設定する</li> </ul>
ビューパッケージの各クラス図には、標準製品仕様書の章節項の番号が名称として付与されている	クラス図名称も取り込み処理を行うため、名称をクラス図名に変更する。
応用スキーマ文書の説明文に該当する情報がない	必要な各オブジェクト（クラス、属性、関連役割等）に対し、説明文を追記する。
ShapeChange で出力した際、関連役割の順番が現行と異なるクラスが発生する	関連役割にプロパティ（sequenceNumber）を追加し、順番を制御する。対象クラスは主として下記である。 _urbanFunction / CensusBlock / DevelopmentProject / TrafficFacility / UrgentUrbanRenewalArea

応用スキーマ文書に表示する説明文は下記のとおり実装するものとする。

- 説明文が必要なオブジェクトの NOTE プロパティに記載する
- 記載方法は MetanormaADOC による直書きとする  
※LutaML 側で MetanormaADOC として取り込み可能にする
- 継承等で同じ説明文が使い回されるため、内容を整理して記載する（現行版と差異が生じる）

### 3-2-2. 符号化仕様作成における開発

EA から XSD を出力するための Config ファイル開発は、既に OGC が CityGML3.0 において導入している（「9. 参考資料」を参照）。

本プロジェクトでは、ShapeChange 本体に付属するテスト用の Config をベースに、OGC の Config を参照して下記の修正を施し、urf と uro それぞれに Config を作成する。

表 3-2 ShapeChange の Config 作成

機能要素	修正
input	<ul style="list-style-type: none"> <li>● インputファイルパスを変更する</li> <li>● "appSchemaName"で urf/uro のいずれかを指定する</li> <li>● "appSchemaNamespaceRegex"はコメントアウトする</li> <li>● &lt;packages&gt;で CityGML2.0 の利用地物や i-UR3.1 の情報を記載する</li> </ul>
targets	<ul style="list-style-type: none"> <li>● OGC の Config を参照し、&lt;rules&gt;を追加する</li> <li>● "StandardNamespaces-v31.xml"をインクルードする</li> <li>● &lt;xmlNamespaces&gt;で CityGML2.0 の利用地物や i-UR3.1 の情報を記載する</li> <li>● 接頭辞が XS や GML で、「接頭辞:名称」が認識できない情報には&lt;xsdMapEntries&gt;を設定する</li> </ul>

※設定内容については ShapeChange 公式 HP の Config 説明を参照のこと（「9.参考資料」を参照）

また ShapeChange 出力された XSD と現行の XSD の相違点、及び整形プログラムにおける対応は下記のとおりとする。なお整形プログラムは ExcelVBA とする。

表 3-3 ShapeChange 出力 XSD に対する整形プログラムの対応

現行の XSD との相違点	整形プログラムの対応
XS 接頭辞が設定・出力されない	XS 接頭辞を設定・出力する。
利用しない情報やデフォルト情報は出力されないため、特にヘッダの形式が変わる、また 1 行目の XML 定義のタグで改行がない	ヘッダで下記を出力する（その他は対応なし） attributeFormDefault="unqualified uro では地形とグループの定義を追加する 1 行目の XML 定義で改行する。
urf/uro の schemaLocation を相対パス指定に変更できない	該当箇所を相対パスで置換する。
substitutionGroup="gml:_Object"の出力がされてしまう	該当プロパティは削除する。
一部のクラスの役割で、Type が Ref に変わることがある	Type に戻す。
「AbstractConstructionType」の役割「boundedBy」が出力できない	挿入する。
クラスの表示順や、クラス内でのプロパティの表示順が、アルファベット順で出力される	プロパティは Name を先頭にする（その他は対応なし）。

現行の XSD との相違点	整形プログラムの対応
urf で「TargetPropertyType」の Ref が、「urf:Target」から変更できない	「core:_CityObject」に置換する。
仕様書上で「core:_CityObjectType」を継承する下記四つのクラスの継承クラスを変更できない AbstractConstruction / ConstructionInstallation _BoundarySurface / _UrbanFunction	「core:AbstractCityObjectType」に置換する。
uro において、substitutionGroup のクラス名に「_GenericApplicationPropertyOf」という文言が含まれているクラスが、EA で定義困難である（利用しているクラスがプロジェクトファイルの範囲をこえるため）	追加する（uro の追加対象クラスは整形プログラムのシート上で、利用者に分かるように明示する）。 なおできるだけ現行 XSD に合わせるため、追加箇所はヘッダ部分の直後とする。
EA 上の「bldg:_AbstractBuildingType」が CityGML では「bldg:AbstractBuildingType」である	置換する。
「AbstractPointCloud」で substitutionGroup を設定できない、また継承を認識できず「fixme:fixme」と表示される	substitutionGroup は「gml:_Feature」、継承は「gml:AbstractFeatureType」に設定する。
列挙子の Name の末尾に文字列「type」が自動付与される	自動付与される「type」を削除する。

※コメントも出力されないが、内容には影響しないため対応しない

### 3-2-3. ドキュメントのスタイル作成における開発

スタイルに対する規格の適用方針は下記のとおりである。

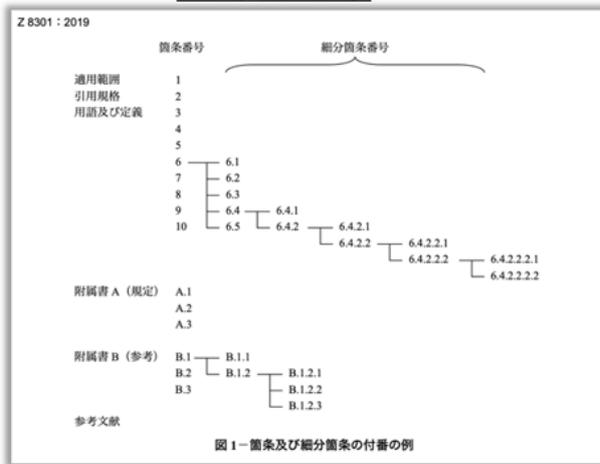
表 3-4 スタイルに対する規格の適用方針

項目	小項目	内容
細分箇条と細別箇条	-	細分箇条は JIS 規格に準じるが、細別箇条は（目次の階層に制限を設けるため）ISO 規格に準じる。（※1） →このため細別箇条に、ISO で利用するアルファベットやエムダッシュ（-）が適用される箇所が生じる。
用語と定義	-	JIS 規格に準じて採番する。
補足 （JIS 対応有）	例、表、図	JIS 規格に準じて採番するが、表の中の図は採番しない。
	式	規格を適用せず、文字列か画像で対応する。
	注	JIS 規格に準じて採番する。 →同じ内容の注は adoc の記載も統一する。
	孤立注	JIS 規格に準じて採番する。 →内容的に注記にそぐわない場合、adoc で通常の文章として扱う。
補足	解説	JIS 規格の注記として扱う。

項目	小項目	内容
(JIS 対応無)	要件、推奨、 留意事項	OGC ModSpec モデル (※2) を利用する。 →「留意事項」は「要件」として扱う。
目次構成		冒頭にある「改定の概要」と、末尾にある「改訂履歴」は結合する。 →結合した「改訂履歴」を冒頭（「はじめに」の後）に配置する。 作業手順書を 0 章始まりから 1 章始まりに変更するとともに、1 章にある節「引用文献」を、標準製品仕様書に合わせて「引用規格等」に変更する。

※1：下図参照

JIS Z 8301 (細分箇条5階層：6.4.2.2.2.1)



細別箇条



図 3-1 細分箇条と細別箇条

※2：下図参照

■登録される要件は各章節の冒頭部に集結させる(省略可能)

表 D-1 - Requirements class D-1: 妥当な交通 (道路) オブジェクト	
Identifier	/req/txan
Target type	道路モデル
Description	妥当な交通 (道路) オブジェクト
Normative statements	Requirement D-1.1: 道路記述のLOD対応条件 Requirement D-1.2: 道路モデルの幾何オブジェクト要件 Requirement D-1.3: txan-Roadの幾何属性一致条件 Requirement D-1.4: txan-Roadの交通禁止条件 Requirement D-1.5: txan-Roadオブジェクトの区切り位置 Requirement D-1.6: txan-Roadの交通属性一致条件 Requirement D-1.7: txan-Roadの交通属性一致条件 Requirement D-1.8: 交通属性の区別方法 Requirement D-1.9: txan-TrafficAreaの裏面禁止条件 Requirement D-1.10: 交通属性の区別方法 Requirement D-1.11: txan-NodeLikeTrafficAreaの裏面禁止条件

D.2 道路の記述とLOD

D.2.1 使用可能な地物型とLOD

CityGMLには、道路を記述するための地物型が用意されている。詳細度 (LOD) ごとに、使用するべき地物型やその空間属性が異なる。

■各要件は表形式で表現される、付属書ごとに接頭辞をつけた採番を行い、冒頭部分とリンクする

表 D-2 - Requirement D-1: 道路記述のLOD対応条件	
Identifier	/req/txan/1
Subject	道路モデル
Included in	Requirements class D-1: 妥当な交通 (道路) オブジェクト
Statement	道路の記述には、指定されたLODに対応する地物型及び空間属性を使用する。

図 3-2 OGC ModSpec の例

また OGC ModSpec の英語列表示は、下記のとおり日本語に置換する。

表 3-5 OGC ModSpec 列表示の日本語置換

英語の列表示	置換する日本語
requirement	要件
recommendation	推奨
permission	許可
obligation	義務
subject	主題
inherits	継承
recommendationtest	推奨検証
requirementtest	要件検証
permissiontest	許可検証
recommendationclass	推奨分類
requirementclass	要件分類
permissionclass	許可分類
abstracttest	抽象検証
conformanceclass	適合分類
conformancetest	適合性検証
targettype	対象の種類
target	対象
testpurpose	検証目的
testmethod	検証方法
dependency	前提条件
indirectdependency	間接的な前提条件
identifier	ID
included_in	分類
statement	条文
description	説明
guidance	ガイダンス
implements	実装
provision	規定

なお表セル内の英字が、列幅に対して長すぎる場合、「-」をいれて折り返すスタイルとなっている。これについてはスタイルが自動で適応の動きをとるため、開発や検証の対象外とする。

uro:largeCustomerFacilityAt-tribute [_AbstractBuilding]	uro:LargeCustomerFacilityAt-tribute [0..*]
--	--

図 3-3 表セル中の英字の折り返し例

### 3-2-4. LutaML のコマンドとレンダリングスタイルの変更

LutaML で開発する応用スキーマ文書のレンダリングコマンド、及びコードリストのレンダリングコマンドとパラメータの仕様は下記のとおりである。

表 3-7 応用スキーマ文書のレンダリングとパラメータ

PT	コマンド	パラメータ内容
1	lutaml_klass_table:: ../../sources/xmi/plateau_all_packages_export.xmi [name="AAA", template="/path/to/templates/_my_klass_table.liquid"]	<ul style="list-style-type: none"> <li>● AAA は「接頭辞::クラス名」で指定する（その他の指定は固定とする）</li> </ul>
2	lutaml_klass_table:: ../../sources/xmi/plateau_all_packages_export.xmi [package="BBB", name="CCC", template="/path/to/templates/_my_klass_table.liquid"]	<ul style="list-style-type: none"> <li>● BBB で接頭辞を指定する</li> <li>● CCC でクラス名を指定する（その他の指定は固定とする）</li> </ul>

※上記 PT1、PT2、いずれでも対応可能である。なお PT3 としてモデル内のパスで指定する方法もあるが、パスが煩雑なので利用しないものとする

表 3-8 コードリストのレンダリング

コマンド	パラメータ内容
lutaml_gml_dictionary::DDD[template="gml_dict_template.liquid", context=dict]	<ul style="list-style-type: none"> <li>● DDD は描画したいコードリスト XML ファイルの相対パスとする（その他の指定は固定とする）</li> </ul>

基本的に XMI で定義されている情報は、出力指定したクラスに関して機械的にすべて応用スキーマ文書に出力されることになるため、適宜省略している現行版よりも、出力が増える見込みである。このため LutaML では下記の対応を行う。

- 定義クラスの情報を、名称の後ろに () で追加して便宜を図る
- 未利用情報への灰色背景色と () 付けは廃止する (OBJ リストに情報あり)

※EA の説明文追記の際に、上記の動きを念頭に置いて記載するものとする

また役割の順序は EA にデータがないため、下記のソートで出力する。

表 3-9 関連役割の出力ソート順

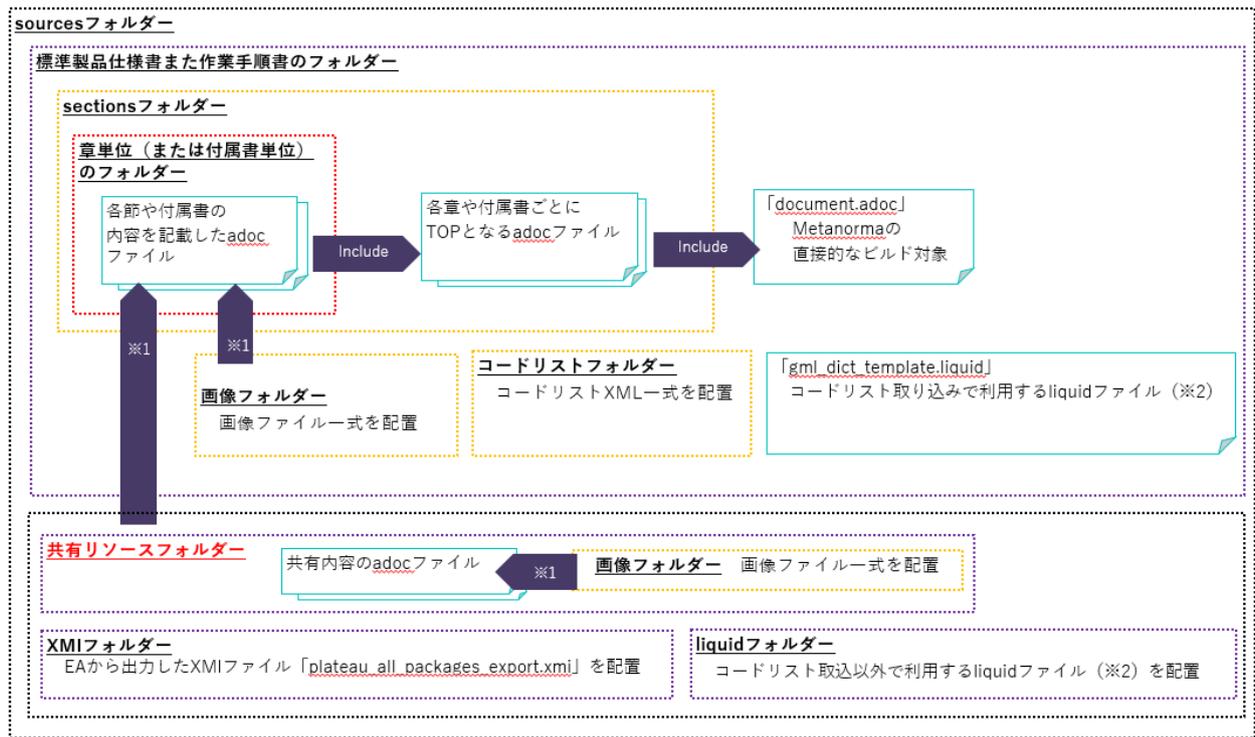
役割	ソート順	ソート項目
継承する関連役割	1	クラスの継承順
	2	接頭辞の優先順 (i-UR 以外の接頭辞>urf>uro)
	3	接頭辞を含めた名称アルファベット順
自身に定義された関連役割	1	接頭辞の優先順 (i-UR 以外の接頭辞>urf>uro)
	2	接頭辞を含めた名称アルファベット順

### 3-2-5. ドキュメントのコンテンツ作成における開発

標準製品仕様書と作業手順書の adoc ファイルその他ドキュメント一式は、下記のフォルダー・ファイル構成とする。

トップフォルダーには「sources」という名称を付与し、サブフォルダーとなる adoc ファイルを構造化して管理するフォルダーには、「sections」という名称を付与する。

なおコードリストについては、Metanorma がファイル生成を行う際、liquid ファイルを通して adoc ファイルに取り込まれ、自動レンダリングされる仕組みである。



- ※1 MetanormaADOCの命令文で取り込みを行う
- ※2 動的にデータを扱いファイルを作成できるliquid構文というスクリプトを記載したファイル

図 3-4 adoc ファイルのインクルード構成

Metanorma がファイル生成する際、「sources」フォルダと同階層にシステムファイルや利用ファイルが必要となる。これらを格納する Metanorma コンテンツ用 TOP フォルダの格納対象は下記のとおりである。

表 3-10 Metanorma コンテンツ用 TOP フォルダの格納対象 (※1)

格納対象	種類	内容
sources	フォルダ	標準製品仕様書と作業手順書の adoc ファイルその他ドキュメント一式を格納する
Gemfile	ファイル	Metanorma の gem ファイル
README.adoc	ファイル	Metanorma plateau のリードミー
collection.yml	ファイル	Metanorma で用いられる YAML (※2) で書かれたシステムファイル
metanorma.yml	ファイル	Metanorma で用いられる YAML (※2) で書かれたシステムファイル
cover.html	ファイル	表紙出力で利用する HTML ファイル

- ※1：参照用フォルダ等、他にも格納物が想定されるが、直接開発内容と関係がないので記載を省略する。
- ※2：JSON と同様、構造化データを表現するデータ形式の一種である。JSON と比べ、人の目での可読性を高めるルールを強いられる。

「sources」フォルダー内で構造化した adoc ファイルに対し、LutaML コマンドを下記の該当箇所で実装する。

- 標準製品仕様書の各地物の応用スキーマ文書を記載する adoc ファイル
- 標準製品仕様書の各地物のコードリストを記載する adoc ファイル

なお「標準製品仕様書の PDF から、AcrobatPro 等で Word 出力する」オペレーションを、フローに追加して行うものとする。

### 3-3. 開発対象の GitHub 管理

3D 都市モデルの GitHub で管理する開発対象は下記とする。

スタイル（3D 都市モデルのフレーバ）と LutaML の新機能については、Metanorma や LutaML の標準レポジトリで管理されるため、対象外とする。

表 3-11 開発対象の GitHub 管理

対象	備考
EA のプロジェクトファイル	旧型式（拡張子 eapx）のファイルである
ShapeChange の Config ファイル	XML 形式のファイルで、urf と uro でファイルが二つある
整形プログラム（ExcelVBA ファイル）	xlsm 形式のファイルである
Metanorma コンテンツ	Metanorma コンテンツ用 TOP フォルダを配置する

## 4. 実証システム

### 4-1. アーキテクチャ

XSD と標準製品仕様書を出力する構成は下図となる。

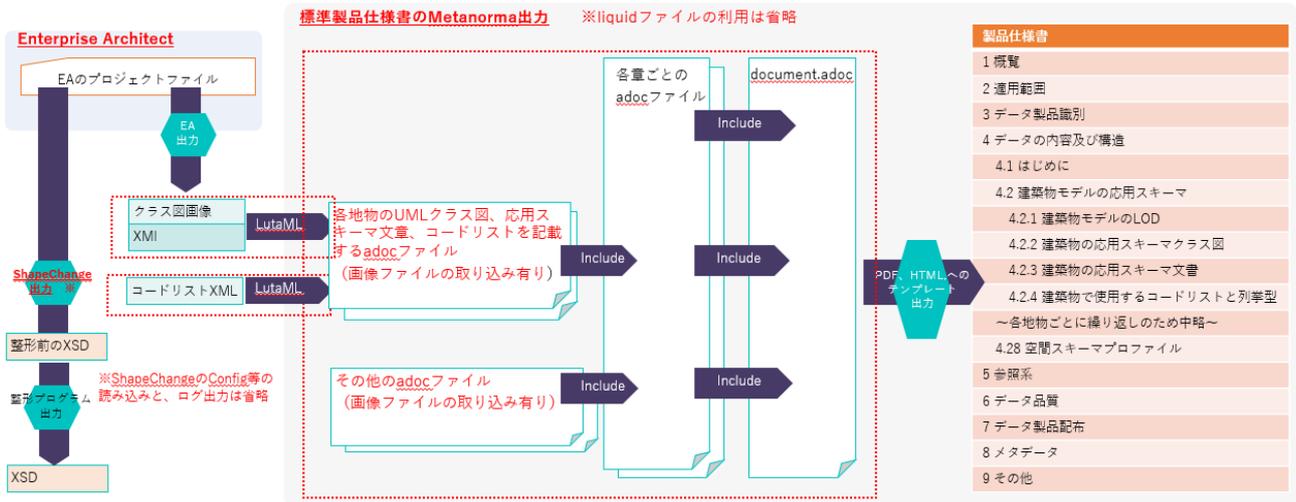


図 4-1 XSD と標準製品仕様書出力の構成

なお、作業手順書を出力する構成は下図となる。

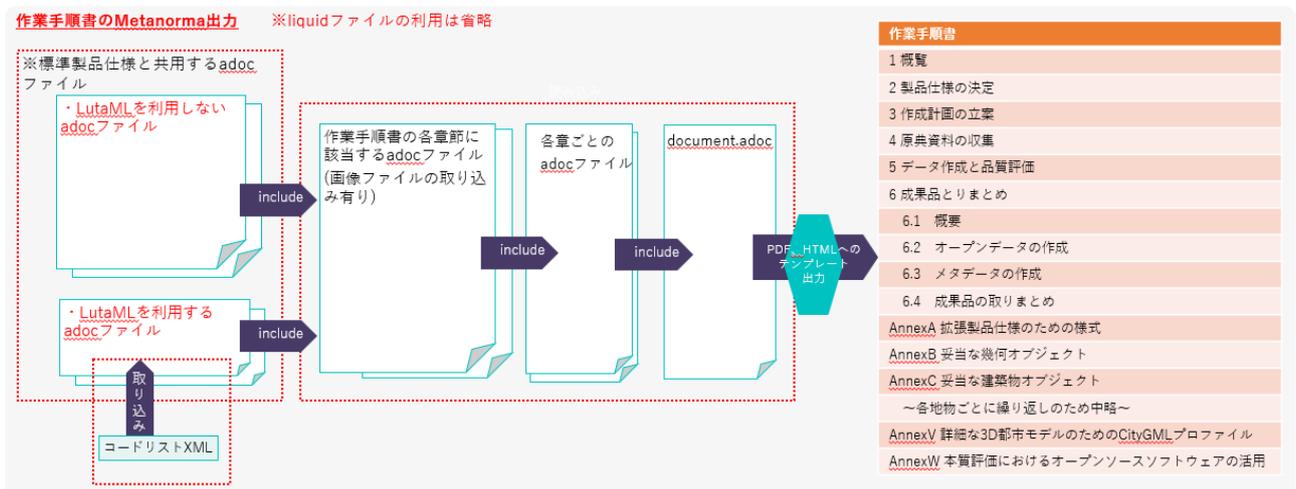


図 4-2 作業手順書出力の構成

## 4-2. 各ソフトウェアの機能と入出力されるファイルフロー

### 4-2-1. 利用ソフトウェアと環境要件

本システムを構築するマシンの OS は Windows11(64bit) とし、20～30G 程度の利用可能なメモリは確保できるものとする。また、本システムを構成するソフトウェアは下記のとおりである。インストールするマシンはソフトウェアごとに分かれていても問題ないが、ShapeChange は EA の環境ファイルを読み込む必要があるため、EA と同じマシンに環境構築する必要がある。

表 4-1 本システムを構成するソフトウェア

ソフトウェア	ライセンスの有無	備考
EA	有	EA は既存フローでも利用しているため、プロジェクトファイルは旧式の EAPX である
ShapeChange	無	バージョン 3.1.0 の JAR ファイルを利用する
Microsoft Office Excel	有	整形プログラムの VBA を動かすために利用する
Metanorma	無	インストールマシンに adoc の「sources」フォルダーを用意する（インストールするマシンは Windows11 を推奨）

ライセンスの必要がないソフトウェアの環境要件は下記のとおりである。なお該当ソフトウェアに対するセキュリティソフト等のチェックは解除しておく必要がある。

※ライセンスの必要があるソフトウェアの要件は、ライセンス購入時の要件とする

表 4-2 フリーソフトウェアの要件

ソフトウェア	要件
ShapeChange	<ul style="list-style-type: none"> <li>● Java の実行環境が構築されていること <ul style="list-style-type: none"> <li>➤ →3.1.0 の JAR ファイルをキックするため、JRE11 の環境を要件とする</li> </ul> </li> <li>● 各種ファイルの読み込みのため、インターネットにアクセス可能であること</li> </ul>
Metanorma	<ul style="list-style-type: none"> <li>● 16GB 以上のメモリのマシンであること</li> <li>● 各種ファイルの読み込みのため、インターネットにアクセス可能であること</li> <li>● （本技術調査では）開発インストールを行う（※1）</li> <li>● Ruby 環境が構築されていること（※2）</li> <li>● Java の実行環境が構築されていること（※3）</li> </ul> <p>→PDF 出力において、依存関係にある JAR ファイルをキックするため</p>

※1：「bundle update」のコマンドで、Ruby の gem ファイルをアップデートが可能であるため

※2：開発キットがあるインストーラでインストールすることを推奨する

※3：Java 環境がない場合、Metanorma インストール時に同時にインストールするものとする。検証では ShapeChange と同じ JRE11 で構築した

## 4-2-2. 機能と入出力されるファイル

各ソフトウェアで利用する機能は下記のとおりである。

表 4-3 各ソフトウェアで利用する機能

No.	ソフトウェア	機能	内容
1	EA	プロジェクトファイル変換	EA のプロジェクトファイル (EAPX) を、ShapeChange で取り込める新形式の QEA に変換出力する
2	EA	モデル・ダイアグラム画像出力	XMI と UML クラス図の画像ファイルを出力する
3	ShapeChange	概念モデルの符号化仕様 (XSD) 変換	Config となる XML ファイルの設定によって、qea ファイルから符号化仕様 (XSD) を出力する
4	整形プログラム (ExcelVBA)	XSD 整形	XSD に対し、「3-2-2. 符号化仕様作成における開発」で示した変更を行う
5	Metanorma	ドキュメント生成	adoc ファイルから、指定した 3D 都市モデルのスタイルで、PDF と HTML を出力する

上記ソフトウェアで入出力されるファイルは下記のとおりである。

なお出力される各種 LOG ファイル及び Config 以外の読み込み設定ファイルや、Metanorma のシステムファイル群等、オペレーションに関係ないファイルは省略する。

表 4-4 各ソフトウェアで利用されるファイル

No.	ファイル	形式・拡張子	単一ファイルか複数ファイルか
1	EAPX ファイル	eapx	単一
2	QEA ファイル	qea	単一
3	ShapeChange の Config	xml	単一
4	整形前の XSD	xsd	複数 (urf と uro でファイルが二つある)
5	整形後の XSD	xsd	複数 (urf と uro でファイルが二つある)
6	XMI ファイル	xmi	単一
7	UML クラス図の画像ファイル	png	複数
8	UML クラス図以外の画像ファイル	(主として) png	複数
9	adoc ファイル	adoc	複数
10	コードリスト	xml	複数
11	生成ドキュメント PDF	pdf	複数 (標準製品仕様書/作業手順書でファイルが二つある)
12	生成ドキュメント HTML	html	複数 (標準製品仕様書/作業手順書でファイルが二つある)

各ソフトウェアで利用する機能において、入出力するファイルは下図のとおりである。

※入出力されるファイル No.は、表 4-4 の No.を利用する

表 4-5 各機能の入出力ファイル

フロー	機能	入力ファイル No.	出力ファイル No.
概念モデルと符号化仕様の作成	プロジェクトファイル変換	1	2
	概念モデルの符号化仕様 (XSD) 変換	2、3	4
	XSD 整形	4	5
概念モデルとドキュメントの生成	モデル・ダイアグラム画像出力	1	6、7
	ドキュメント生成	6、7、8、9、10	11、12

### 4-2-3. データフロー

「4-2-2. 機能と入出力されるファイル」で示した各要素の関係を、下記のフロー図において示す。

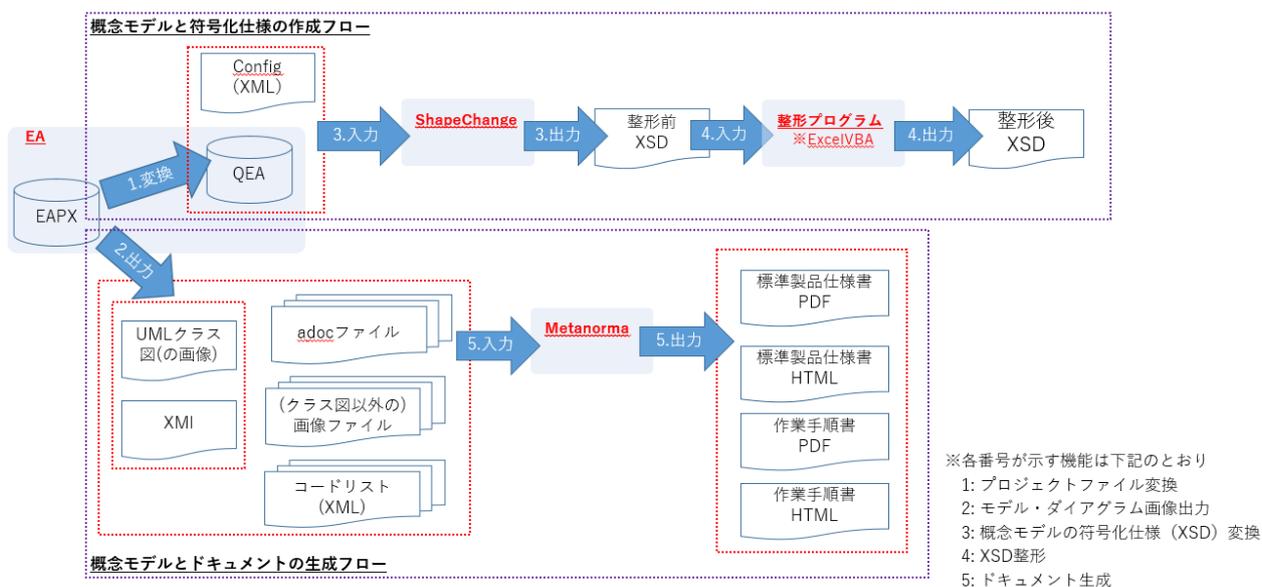


図 4-3 データフロー

### 4-3. 各機能の利用手順

#### 4-3-1. EA のプロジェクトファイル変換

EA のプロジェクトファイル変換は下記の手順である。

1. 「メインメニュー」の「プロジェクト」の「転送」から、「EAP/EAPX ファイルの変換」を選択する
2. サブウィンドウにおいて、インプットをもとのプロジェクトファイル、出力を「QEA ファイル」で任意の名前を設定し、転送する



図 4-4 EA のプロジェクトファイル変換

#### 4-3-2. EA のモデル・ダイアグラム画像出力

EA のプロジェクトファイル変換は下記の手順である。

1. 「メインメニュー」の「生成・入出力」の「出力」から、「その他の形式」を選択する
2. サブウィンドウにおいて下表の出力設定を行い、「出力」ボタンを押下する

なおダイアグラムの画像フォーマットで、SVG と PNG のそれぞれを設定して、二回出力を行う。

表 4-6 EA のモデル・ダイアグラム画像出力設定

設定項目	設定内容
パッケージ	トップのパッケージを設定する
ファイル名	出力 XMI パスを設定する ファイル名は「plateau_all_packages_export.xmi」とする
出力形式	UML2.2.1 (XMI2.5.1) を選択する
オプション	UML2.2.1 (XMI2.5.1) のデフォルト選択オプションに加え、「ダイアグラムの画像を生成」を選択し、ダイアグラムを出力する ※デフォルト選択オプションは下記のとおりである ダイアグラムを出力 / 整形して出力 / DTD の利用
ダイアグラムの画像フォーマット	SVG と PNG をそれぞれ選択する



図 4-5 EA のモデル・ダイアグラム画像出力

### 4-3-3. ShapeChange の XSD 出力

ShapeChange による XSD 出力は下記の手順である。

1. PowerShell を管理者モードで開き、カレントディレクトリを ShapeChange の JAR ファイルのあるフォルダーに変更する
2. 「java -jar ShapeChange-3.1.0.jar -Dfile.encoding=UTF-8 -c AAA」を実行する。AAA は Config ファイルのパスであり、Config 指定の出力先に XSD が出力される

### 4-3-4. ExcelVBA による XSD 整形

ExcelVBA による XSD 整形は下記の手順である。

1. 対象 xlsx ファイルのワークシート「do」の表に、urf/uro の入出力 XSD 及び uro のバージョンを小数点以下一位まで設定する
2. ワークシート「uro\_add」に、uro で追加したい行を設定する
3. 「実行ボタン」を押下する

	SC出力	整形後出力
urf	C:\Users\501829\Desktop\新しいフォルダー (2)\urbanFunction_ShapeChange.xsd	C:\Users\501829\Desktop\新しいフォルダー (2)\整形後\urbanFunction_ShapeChange.xsd
uro	C:\Users\501829\Desktop\新しいフォルダー (2)\urbanObject_ShapeChange.xsd	C:\Users\501829\Desktop\新しいフォルダー (2)\整形後\urbanObject_ShapeChange.xsd
URO-ver	3.2	実行ボタン

図 4-6 ワークシート「do」



図 4-7 ワークシート「uro\_add」

### 4-3-5. Metanorma によるドキュメント生成

Metanorma によるドキュメント生成は下記の手順である。

1. PowerShell を管理者モードで開き、カレントディレクトリを、「Metanorma コンテンツ用 TOP フォルダ」に変更する
2. カレントディレクトリから「Gemfile.lock」を削除する
3. 「bundle install」と「bundle update」を実行する
4. 下表のとおりコマンドを実行する

表 4-7 Metanorma によるドキュメントコマンド

生成対象	コマンド	出力先
標準製品仕様書	bundle exec metanorma sources/001-v5/document.adoc	001-v5
作業手順書	bundle exec metanorma sources/002-v5/document.adoc	002-v5
標準製品仕様書 と作業手順書	bundle exec metanorma site generate --agree-to-terms	「Metanorma コンテンツ用 TOP フォルダ」直下に出カ フォルダが生成される

## 5. 品質

---

「概念モデルと符号化仕様の作成」と「概念モデルとドキュメントの生成」のフローの処理が最後まで実行されることを前提とし、「4. 実証システム」で示した各機能の機能要件及び非機能要件の品質担保を下記のとおり行う。

なお「プロジェクトファイル変換」と「モデル・ダイアグラム画像出力」の機能については、EA の標準機能を用いるため対象外とする。

「符号化仕様作成」は ShapeChange の標準機能であるため、非機能要件については対象外とするが、開発対象である Config ファイルによって動きが変わるため、機能要件は対象とする。

「ドキュメント生成」は Metamora の標準機能であるが、スタイル・コンテンツ共に開発対象であり、3D 都市モデルのドキュメント生成としては標準機能で品質担保はできないため、対象外とはしない。

## 5-1. 機能要件の品質担保

品質を担保すべき機能要件を、機能のフローごとに以下に示す。

「概念モデルと符号化仕様の作成」の機能要件は、下表のとおりである。

表 5-1 「概念モデルと符号化仕様の作成」の機能要件

機能	評価項目	目標
概念モデルの符号化仕様 (XSD) 変換	<ul style="list-style-type: none"> <li>EA や ShapeChange の Config に反映した修正が、XSD にも反映される</li> </ul>	<ul style="list-style-type: none"> <li>Config に設定した&lt;xsdMapEntries&gt;の指定どおり出力される</li> <li>役割が EA で定義した順番どおり出力される</li> </ul>
XSD 整形	<ul style="list-style-type: none"> <li>現行 XSD と同内容かつ可能な限り同形式の XSD を出力できる</li> </ul>	<ul style="list-style-type: none"> <li>XS 接頭辞が設定される</li> <li>現行 XSD と同内容のヘッダが記載される</li> <li>REF に変わった Type をもとに戻す</li> <li>名称末尾に Type がついた列挙子の名称をもとに戻す</li> <li>「AbstractConstructionType」の役割に「boundedBy」がある</li> <li>プロパティは Name を先頭にする</li> <li>「TargetPropertyType」の Ref が「core:_CityObject」である</li> <li>substitutionGroup="gml:_Object"は出力されない</li> <li>「core:_CityObjectType」を継承するクラスの継承クラスは「core:AbstractCityObjectType」に置換される</li> <li>ヘッダ部分の直後に、uro 追加対象が追加されている</li> <li>「bldg:AbstractBuildingType」が「bldg:_AbstractBuildingType」に置換されている</li> <li>「AbstractPointCloud」の継承は「gml:AbstractFeatureType」に、substitutionGroup は「gml:_Feature」に設定する</li> </ul>

「概念モデルとドキュメントの生成」の機能要件は、下表のとおりである。

表 5-2 「概念モデルとドキュメントの生成」の機能要件

機能	評価項目	目標
ドキュメント生成	<ul style="list-style-type: none"> <li>● 開発したスタイルの定義が反映されている</li> </ul>	<ul style="list-style-type: none"> <li>● 表の中の注は自動採番されない</li> <li>● OGC ModSpec が正常に表示されている</li> <li>● 3-2 で示した目次構成変更が反映されている</li> </ul>
	<ul style="list-style-type: none"> <li>● 応用スキーマ文書とコードリストが現行ドキュメントと可能な限り同内容かつ同スタイルで作成される</li> </ul>	<ul style="list-style-type: none"> <li>● 自動レンダリングされる応用スキーマ文書で、EA で定義した説明文が出力され、定義もとのクラス情報が記載される</li> <li>● 応用スキーマ文書における関連役割の出力順が、「3-2. 開発内容」と相違なく出力される</li> <li>● コードリストが自動レンダリングされる</li> </ul>
	<ul style="list-style-type: none"> <li>● 現行ドキュメントと同内容かつ可能な限り同スタイルで出力できる</li> </ul>	<ul style="list-style-type: none"> <li>● UML クラス図のファイルタイトルが EA の定義と相違なく表示される</li> </ul>
	<ul style="list-style-type: none"> <li>● EA に反映した修正が、ドキュメントにも反映される</li> </ul>	<ul style="list-style-type: none"> <li>● 標準製品仕様書と作業手順書がともに出力可能で、共有リソースへの修正が双方に同内容で反映される</li> </ul>

## 5-2. 非機能要件の品質担保

品質を担保すべき非機能要件を、機能のフローごとに以下に示す。

表 5-3 「概念モデルと符号化仕様の作成」の非機能要件

機能	評価項目	目標
XSD 整形	● ドキュメント生成のオペレーションが容易に実行可能である	● パラメータ設定とボタンの押下のみで実行可能である
	● ドキュメント生成の速度が極端に遅くない	● 30 秒以内で処理完了とする

表 5-4 「概念モデルとドキュメントの生成」の非機能要件

機能	評価項目	目標
ドキュメント生成	● ドキュメント生成のオペレーションが容易に実行可能である	● ファイル生成自体は、1 ファイルに対し 1 コマンドで実行可能であること
	● ドキュメント出力の速度が極端に遅くない	● 標準製品仕様書は 1~2 時間前後、作業手順書は 30 分前後で出力可能とする (※1)

※1：標準製品仕様書は LutaML を多用するため、作業手順書より大幅に処理時間を要する。また実行時間は通信環境に大きく影響される

## 6. 実証技術の機能要件の品質検証

### 6-1. 概念モデルと符号化仕様の作成検証

#### 6-1-1. 検証方法

下記のとおり「概念モデルの符号化仕様（XSD）変換」と「XSD 整形」機能の機能要件を検証した。

表 6-1 XSD 変換と XSD 整形の機能要件検証

機能	No.	評価目標	検証方法
XSD 変換	1	Config に設定した<xsdMapEntries>の指定どおり出力されている	<xsdMapEntries>に設定したデータ型に対し、EA の設定と XSD の出力を対比して確認する。
	2	役割が EA で定義した順番どおり出力されている	順番設定した EA の設定と XSD の出力を対比して確認する。
XSD 整形	1	XS 接頭辞が設定される	XSD 整形前後でファイルを比較して確認する。
	2	REF に変わった Type をもとに戻す	
	3	プロパティは Name を先頭にする	
	4	「TargetPropertyType」の Ref は「core:_CityObject」とする	
	5	「core:_CityObjectType」を継承するクラスの継承クラスは、「core:AbstractCityObjectType」に置換される	
	6	「AbstractConstructionType」に役割「boundedBy」を挿入する	
	7	「bldg:_AbstractBuildingType」を「bldg:AbstractBuildingType」に置換する	
	8	名称末尾に Type がついた列挙子の名称をもとに戻す	
	9	現行 XSD と同内容のヘッダが記載される	
	10	「AbstractPointCloud」の substitutionGroup は「gml:_Feature」、継承は「gml:AbstractFeatureType」に設定する	
	11	substitutionGroup="gml:_Object"は出力されない	整形後 XSD を検索し、出力されていないことを確認する。
		12	ヘッダ部分の直後に追加対象クラスが追加される

※1：出力前後にコメントを書き込み確認可能とする

## 6-1-2. 検証結果 (XSD 変換)

1) <xsdMapEntries>に設定したデータ型に対し、EA の設定と XSD の出力を対比して確認  
全クラスを確認するのは困難であるため、ここでは「\_urbanfunction」の属性を例に確認する。

「\_urbanfunction」の属性について、EA での定義は下図のとおりである。

名前	種類	名前	種類
class	gml:CodeType	finalNotificationDate	xs:date
function	gml:CodeType	finalNotificationNumber	xs:string
usage	gml:CodeType	urbanPlanType	gml:CodeType
validFrom	xs:date	areaClassificationType	gml:CodeType
validFromType	gml:CodeType	nominalArea	gml:MeasureYear
enactmentFiscalYear	xs:gYear	prefecture	gml:CodeType
validTo	xs:date	city	gml:CodeType
validToType	gml:CodeType	reference	xs:anyURI
expirationFiscalYear	xs:gYear	reason	gml:StringOrRefType
legalGrounds	xs:string	note	gml:StringOrRefType
custodian	xs:string	surveyYear	xs:gYear
notificationNumber	xs:string		

図 6-1 EA における「\_urbanfunction」の属性型定義

ShapeChange から出力した XSD では下図のとおり、EA 上での属性の型における種類の接頭辞「xs::」と「gml::」が、「xs:」と「gml:」に変換されている。

```

<complexType abstract="true" name="_UrbanFunctionType">
  <complexContent>
    <extension base="core:_CityObjectType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType"/>
        <element minOccurs="0" name="usage" type="gml:CodeType"/>
        <element minOccurs="0" name="validFrom" type="xs:date"/>
        <element minOccurs="0" name="validFromType" type="gml:CodeType"/>
        <element minOccurs="0" name="enactmentFiscalYear" type="xs:gYear"/>
        <element minOccurs="0" name="validTo" type="xs:date"/>
        <element minOccurs="0" name="validToType" type="gml:CodeType"/>
        <element minOccurs="0" name="expirationFiscalYear" type="xs:gYear"/>
        <element minOccurs="0" name="legalGrounds" type="xs:string"/>
        <element minOccurs="0" name="custodian" type="xs:string"/>
        <element minOccurs="0" name="notificationNumber" type="xs:string"/>
        <element minOccurs="0" name="finalNotificationDate" type="xs:date"/>
        <element minOccurs="0" name="finalNotificationNumber" type="xs:string"/>
        <element minOccurs="0" name="urbanPlanType" type="gml:CodeType"/>
        <element minOccurs="0" name="areaClassificationType" type="gml:CodeType"/>
        <element minOccurs="0" name="nominalArea" type="gml:MeasureType"/>
        <element minOccurs="0" name="prefecture" type="gml:CodeType"/>
        <element minOccurs="0" name="city" type="gml:CodeType"/>
        <element minOccurs="0" name="reference" type="xs:anyURI"/>
        <element minOccurs="0" name="reason" type="gml:StringOrRefType"/>
        <element minOccurs="0" name="note" type="gml:StringOrRefType"/>
        <element minOccurs="0" name="surveyYear" type="xs:gYear"/>
        <element minOccurs="0" name="lod-2MultiSurface" type="gml:MultiSurfacePropertyType"/>
        <element minOccurs="0" name="lod-1MultiSurface" type="gml:MultiSurfacePropertyType"/>
        <element minOccurs="0" name="lod0MultiSurface" type="gml:MultiSurfacePropertyType"/>
        <element minOccurs="0" name="lod-2MultiCurve" type="gml:MultiCurvePropertyType"/>
        <element minOccurs="0" name="lod-1MultiCurve" type="gml:MultiCurvePropertyType"/>
        <element minOccurs="0" name="lod0MultiCurve" type="gml:MultiCurvePropertyType"/>
        <element minOccurs="0" name="lod-2MultiPoint" type="gml:MultiPointPropertyType"/>
        <element minOccurs="0" name="lod-1MultiPoint" type="gml:MultiPointPropertyType"/>
        <element minOccurs="0" name="lod0MultiPoint" type="gml:MultiPointPropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="target"/>
      </sequence>
      <element ref="core:_CityObject"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="UrbanFunctionPropertyType">
  <sequence minOccurs="0">
    <element ref="urf:_UrbanFunction"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

図 6-2 ShapeChange による「\_urbanfunction」の出力

2) 順番設定した EA の設定と XSD の出力を対比して確認

また図 6-2 の関連役割の順番は、図 6-3~6-5 に示す EA の順番設定と同じであることが確認できる。

<p>ターゲット: MultiSurface</p> <p><u>lod-2MultiSurface</u></p> <hr/> <p>多重度</p> <p>多重度 0..1</p> <p>Ordered False</p> <p>重複を許可 False</p> <p>詳細</p> <p>ステレオタイプ GML::property</p> <p>別名</p> <p>可視性 Public</p> <p>誘導可能性 誘導可能</p> <p>集約 なし</p> <p>スコープ インスタンス</p> <p>制約</p> <p>限定子</p> <p>「property」 (定義: GML)</p> <p>sequenceNumber <b>1</b></p> <p>inlineOrByReference inline</p> <p>isMetadata false</p> <p>追加設定</p>	<p>ターゲット: MultiSurface</p> <p><u>lod-1MultiSurface</u></p> <hr/> <p>多重度</p> <p>多重度 0..1</p> <p>Ordered False</p> <p>重複を許可 False</p> <p>詳細</p> <p>ステレオタイプ GML::property</p> <p>別名</p> <p>可視性 Public</p> <p>誘導可能性 誘導可能</p> <p>集約 なし</p> <p>スコープ インスタンス</p> <p>制約</p> <p>限定子</p> <p>「property」 (定義: GML)</p> <p>sequenceNumber <b>2</b></p> <p>inlineOrByReference inline</p> <p>isMetadata false</p> <p>追加設定</p>	<p>ターゲット: MultiSurface</p> <p><u>lod0MultiSurface</u></p> <hr/> <p>多重度</p> <p>多重度 0..1</p> <p>Ordered False</p> <p>重複を許可 False</p> <p>詳細</p> <p>ステレオタイプ GML::property</p> <p>別名</p> <p>可視性 Public</p> <p>誘導可能性 誘導可能</p> <p>集約 なし</p> <p>スコープ インスタンス</p> <p>制約</p> <p>限定子</p> <p>「property」 (定義: GML)</p> <p>sequenceNumber <b>3</b></p> <p>inlineOrByReference inline</p> <p>isMetadata false</p> <p>追加設定</p>
<p>ソース: MultiSurface</p> <p><u>lod1MultiSurface</u></p> <hr/> <p>多重度</p> <p>多重度 0..1</p> <p>Ordered False</p> <p>重複を許可 False</p> <p>詳細</p> <p>ステレオタイプ GML::property</p> <p>別名</p> <p>可視性 Public</p> <p>誘導可能性 誘導可能</p> <p>集約 なし</p> <p>スコープ インスタンス</p> <p>制約</p> <p>限定子</p> <p>「property」 (定義: GML)</p> <p>sequenceNumber <b>4</b></p> <p>inlineOrByReference inline</p> <p>isMetadata false</p> <p>追加設定</p>	<p>ソース: MultiCurve</p> <p><u>lod-2MultiCurve</u></p> <hr/> <p>多重度</p> <p>多重度 0..1</p> <p>Ordered False</p> <p>重複を許可 False</p> <p>詳細</p> <p>ステレオタイプ GML::property</p> <p>別名</p> <p>可視性 Public</p> <p>誘導可能性 誘導可能</p> <p>集約 なし</p> <p>スコープ インスタンス</p> <p>制約</p> <p>限定子</p> <p>「property」 (定義: GML)</p> <p>sequenceNumber <b>5</b></p> <p>inlineOrByReference inline</p> <p>isMetadata false</p> <p>追加設定</p>	<p>ソース: MultiCurve</p> <p><u>lod-1MultiCurve</u></p> <hr/> <p>多重度</p> <p>多重度 0..1</p> <p>Ordered False</p> <p>重複を許可 False</p> <p>詳細</p> <p>ステレオタイプ GML::property</p> <p>別名</p> <p>可視性 Public</p> <p>誘導可能性 誘導可能</p> <p>集約 なし</p> <p>スコープ インスタンス</p> <p>制約</p> <p>限定子</p> <p>「property」 (定義: GML)</p> <p>sequenceNumber <b>6</b></p> <p>inlineOrByReference inline</p> <p>isMetadata false</p> <p>追加設定</p>

図 6-3 EA における「\_urbanfunction」の関連役割順番定義①

<p>ターゲット: MultiCurve</p> <p><u>lod0MultiCurve</u></p> <hr/> <p>多重度</p> <table border="1"> <tr><td>多重度</td><td>0..1</td></tr> <tr><td>Ordered</td><td>False</td></tr> <tr><td>重複を許可</td><td>False</td></tr> </table> <p>詳細</p> <table border="1"> <tr><td>ステレオタイプ</td><td>GML::property</td></tr> <tr><td>別名</td><td></td></tr> <tr><td>可視性</td><td>Public</td></tr> <tr><td>誘導可能性</td><td>誘導可能</td></tr> <tr><td>集約</td><td>なし</td></tr> <tr><td>スコープ</td><td>インスタンス</td></tr> <tr><td>制約</td><td></td></tr> <tr><td>限定子</td><td></td></tr> </table> <p>«property» (定義: GML)</p> <table border="1"> <tr><td>sequenceNumber</td><td>7</td></tr> <tr><td>inlineOrByReference</td><td>inline</td></tr> <tr><td>isMetadata</td><td>false</td></tr> </table> <p>追加設定</p>	多重度	0..1	Ordered	False	重複を許可	False	ステレオタイプ	GML::property	別名		可視性	Public	誘導可能性	誘導可能	集約	なし	スコープ	インスタンス	制約		限定子		sequenceNumber	7	inlineOrByReference	inline	isMetadata	false	<p>ソース: MultiCurve</p> <p><u>lod1MultiCurve</u></p> <hr/> <p>多重度</p> <table border="1"> <tr><td>多重度</td><td>0..1</td></tr> <tr><td>Ordered</td><td>False</td></tr> <tr><td>重複を許可</td><td>False</td></tr> </table> <p>詳細</p> <table border="1"> <tr><td>ステレオタイプ</td><td>GML::property</td></tr> <tr><td>別名</td><td></td></tr> <tr><td>可視性</td><td>Public</td></tr> <tr><td>誘導可能性</td><td>誘導可能</td></tr> <tr><td>集約</td><td>なし</td></tr> <tr><td>スコープ</td><td>インスタンス</td></tr> <tr><td>制約</td><td></td></tr> <tr><td>限定子</td><td></td></tr> </table> <p>«property» (定義: GML)</p> <table border="1"> <tr><td>sequenceNumber</td><td>8</td></tr> <tr><td>inlineOrByReference</td><td>inline</td></tr> <tr><td>isMetadata</td><td>false</td></tr> </table> <p>追加設定</p>	多重度	0..1	Ordered	False	重複を許可	False	ステレオタイプ	GML::property	別名		可視性	Public	誘導可能性	誘導可能	集約	なし	スコープ	インスタンス	制約		限定子		sequenceNumber	8	inlineOrByReference	inline	isMetadata	false	<p>ターゲット: MultiPoint</p> <p><u>lod-2MultiPoint</u></p> <hr/> <p>多重度</p> <table border="1"> <tr><td>多重度</td><td>0..1</td></tr> <tr><td>Ordered</td><td>False</td></tr> <tr><td>重複を許可</td><td>False</td></tr> </table> <p>詳細</p> <table border="1"> <tr><td>ステレオタイプ</td><td>GML::property</td></tr> <tr><td>別名</td><td></td></tr> <tr><td>可視性</td><td>Public</td></tr> <tr><td>誘導可能性</td><td>誘導可能</td></tr> <tr><td>集約</td><td>なし</td></tr> <tr><td>スコープ</td><td>インスタンス</td></tr> <tr><td>制約</td><td></td></tr> <tr><td>限定子</td><td></td></tr> </table> <p>«property» (定義: GML)</p> <table border="1"> <tr><td>sequenceNumber</td><td>9</td></tr> <tr><td>inlineOrByReference</td><td>inline</td></tr> <tr><td>isMetadata</td><td>false</td></tr> </table> <p>追加設定</p>	多重度	0..1	Ordered	False	重複を許可	False	ステレオタイプ	GML::property	別名		可視性	Public	誘導可能性	誘導可能	集約	なし	スコープ	インスタンス	制約		限定子		sequenceNumber	9	inlineOrByReference	inline	isMetadata	false
多重度	0..1																																																																																					
Ordered	False																																																																																					
重複を許可	False																																																																																					
ステレオタイプ	GML::property																																																																																					
別名																																																																																						
可視性	Public																																																																																					
誘導可能性	誘導可能																																																																																					
集約	なし																																																																																					
スコープ	インスタンス																																																																																					
制約																																																																																						
限定子																																																																																						
sequenceNumber	7																																																																																					
inlineOrByReference	inline																																																																																					
isMetadata	false																																																																																					
多重度	0..1																																																																																					
Ordered	False																																																																																					
重複を許可	False																																																																																					
ステレオタイプ	GML::property																																																																																					
別名																																																																																						
可視性	Public																																																																																					
誘導可能性	誘導可能																																																																																					
集約	なし																																																																																					
スコープ	インスタンス																																																																																					
制約																																																																																						
限定子																																																																																						
sequenceNumber	8																																																																																					
inlineOrByReference	inline																																																																																					
isMetadata	false																																																																																					
多重度	0..1																																																																																					
Ordered	False																																																																																					
重複を許可	False																																																																																					
ステレオタイプ	GML::property																																																																																					
別名																																																																																						
可視性	Public																																																																																					
誘導可能性	誘導可能																																																																																					
集約	なし																																																																																					
スコープ	インスタンス																																																																																					
制約																																																																																						
限定子																																																																																						
sequenceNumber	9																																																																																					
inlineOrByReference	inline																																																																																					
isMetadata	false																																																																																					
<p>ターゲット: MultiPoint</p> <p><u>lod-1MultiPoint</u></p> <hr/> <p>多重度</p> <table border="1"> <tr><td>多重度</td><td>0..1</td></tr> <tr><td>Ordered</td><td>False</td></tr> <tr><td>重複を許可</td><td>False</td></tr> </table> <p>詳細</p> <table border="1"> <tr><td>ステレオタイプ</td><td>GML::property</td></tr> <tr><td>別名</td><td></td></tr> <tr><td>可視性</td><td>Public</td></tr> <tr><td>誘導可能性</td><td>誘導可能</td></tr> <tr><td>集約</td><td>なし</td></tr> <tr><td>スコープ</td><td>インスタンス</td></tr> <tr><td>制約</td><td></td></tr> <tr><td>限定子</td><td></td></tr> </table> <p>«property» (定義: GML)</p> <table border="1"> <tr><td>sequenceNumber</td><td>10</td></tr> <tr><td>inlineOrByReference</td><td>inline</td></tr> <tr><td>isMetadata</td><td>false</td></tr> </table> <p>追加設定</p>	多重度	0..1	Ordered	False	重複を許可	False	ステレオタイプ	GML::property	別名		可視性	Public	誘導可能性	誘導可能	集約	なし	スコープ	インスタンス	制約		限定子		sequenceNumber	10	inlineOrByReference	inline	isMetadata	false	<p>ターゲット: MultiPoint</p> <p><u>lod0MultiPoint</u></p> <hr/> <p>多重度</p> <table border="1"> <tr><td>多重度</td><td>0..1</td></tr> <tr><td>Ordered</td><td>False</td></tr> <tr><td>重複を許可</td><td>False</td></tr> </table> <p>詳細</p> <table border="1"> <tr><td>ステレオタイプ</td><td>GML::property</td></tr> <tr><td>別名</td><td></td></tr> <tr><td>可視性</td><td>Public</td></tr> <tr><td>誘導可能性</td><td>誘導可能</td></tr> <tr><td>集約</td><td>なし</td></tr> <tr><td>スコープ</td><td>インスタンス</td></tr> <tr><td>制約</td><td></td></tr> <tr><td>限定子</td><td></td></tr> </table> <p>«property» (定義: GML)</p> <table border="1"> <tr><td>sequenceNumber</td><td>11</td></tr> <tr><td>inlineOrByReference</td><td>inline</td></tr> <tr><td>isMetadata</td><td>false</td></tr> </table> <p>追加設定</p>	多重度	0..1	Ordered	False	重複を許可	False	ステレオタイプ	GML::property	別名		可視性	Public	誘導可能性	誘導可能	集約	なし	スコープ	インスタンス	制約		限定子		sequenceNumber	11	inlineOrByReference	inline	isMetadata	false	<p>ターゲット: _CityObject</p> <p><u>target</u></p> <hr/> <p>多重度</p> <table border="1"> <tr><td>多重度</td><td>0..*</td></tr> <tr><td>Ordered</td><td>False</td></tr> <tr><td>重複を許可</td><td>False</td></tr> </table> <p>詳細</p> <table border="1"> <tr><td>ステレオタイプ</td><td>GML::property</td></tr> <tr><td>別名</td><td></td></tr> <tr><td>可視性</td><td>Public</td></tr> <tr><td>誘導可能性</td><td>誘導可能</td></tr> <tr><td>集約</td><td>なし</td></tr> <tr><td>スコープ</td><td>インスタンス</td></tr> <tr><td>制約</td><td></td></tr> <tr><td>限定子</td><td></td></tr> </table> <p>«property» (定義: GML)</p> <table border="1"> <tr><td>sequenceNumber</td><td>12</td></tr> <tr><td>inlineOrByReference</td><td>inline</td></tr> <tr><td>isMetadata</td><td>false</td></tr> </table> <p>追加設定</p>	多重度	0..*	Ordered	False	重複を許可	False	ステレオタイプ	GML::property	別名		可視性	Public	誘導可能性	誘導可能	集約	なし	スコープ	インスタンス	制約		限定子		sequenceNumber	12	inlineOrByReference	inline	isMetadata	false
多重度	0..1																																																																																					
Ordered	False																																																																																					
重複を許可	False																																																																																					
ステレオタイプ	GML::property																																																																																					
別名																																																																																						
可視性	Public																																																																																					
誘導可能性	誘導可能																																																																																					
集約	なし																																																																																					
スコープ	インスタンス																																																																																					
制約																																																																																						
限定子																																																																																						
sequenceNumber	10																																																																																					
inlineOrByReference	inline																																																																																					
isMetadata	false																																																																																					
多重度	0..1																																																																																					
Ordered	False																																																																																					
重複を許可	False																																																																																					
ステレオタイプ	GML::property																																																																																					
別名																																																																																						
可視性	Public																																																																																					
誘導可能性	誘導可能																																																																																					
集約	なし																																																																																					
スコープ	インスタンス																																																																																					
制約																																																																																						
限定子																																																																																						
sequenceNumber	11																																																																																					
inlineOrByReference	inline																																																																																					
isMetadata	false																																																																																					
多重度	0..*																																																																																					
Ordered	False																																																																																					
重複を許可	False																																																																																					
ステレオタイプ	GML::property																																																																																					
別名																																																																																						
可視性	Public																																																																																					
誘導可能性	誘導可能																																																																																					
集約	なし																																																																																					
スコープ	インスタンス																																																																																					
制約																																																																																						
限定子																																																																																						
sequenceNumber	12																																																																																					
inlineOrByReference	inline																																																																																					
isMetadata	false																																																																																					

図 6-4 EA における「\_urbanfunction」の関連役割順番定義②

ソース: KeyValuePairAttribute	ソース: DataQualityAttribute																																																																								
<u>KeyValuePairAttribute</u>	<u>dataQualityAttribute</u>																																																																								
<table border="1"> <tr> <td colspan="2">多重度</td> </tr> <tr> <td>多重度</td> <td>0..*</td> </tr> <tr> <td>Ordered</td> <td>False</td> </tr> <tr> <td>重複を許可</td> <td>False</td> </tr> <tr> <td colspan="2">詳細</td> </tr> <tr> <td>ステレオタイプ</td> <td>GML::property</td> </tr> <tr> <td>別名</td> <td></td> </tr> <tr> <td>可視性</td> <td>Public</td> </tr> <tr> <td>誘導可能性</td> <td>誘導可能</td> </tr> <tr> <td>集約</td> <td>なし</td> </tr> <tr> <td>スコープ</td> <td>インスタンス</td> </tr> <tr> <td>制約</td> <td></td> </tr> <tr> <td>限定子</td> <td></td> </tr> <tr> <td colspan="2">«property» (定義: GML)</td> </tr> <tr> <td>sequenceNumber</td> <td>13</td> </tr> <tr> <td>inlineOrByReference</td> <td>inline</td> </tr> <tr> <td>isMetadata</td> <td>false</td> </tr> <tr> <td colspan="2">追加設定</td> </tr> </table>	多重度		多重度	0..*	Ordered	False	重複を許可	False	詳細		ステレオタイプ	GML::property	別名		可視性	Public	誘導可能性	誘導可能	集約	なし	スコープ	インスタンス	制約		限定子		«property» (定義: GML)		sequenceNumber	13	inlineOrByReference	inline	isMetadata	false	追加設定		<table border="1"> <tr> <td colspan="2">多重度</td> </tr> <tr> <td>多重度</td> <td>0..1</td> </tr> <tr> <td>Ordered</td> <td>False</td> </tr> <tr> <td>重複を許可</td> <td>False</td> </tr> <tr> <td colspan="2">詳細</td> </tr> <tr> <td>ステレオタイプ</td> <td>GML::property</td> </tr> <tr> <td>別名</td> <td></td> </tr> <tr> <td>可視性</td> <td>Public</td> </tr> <tr> <td>誘導可能性</td> <td>誘導可能</td> </tr> <tr> <td>集約</td> <td>なし</td> </tr> <tr> <td>スコープ</td> <td>インスタンス</td> </tr> <tr> <td>制約</td> <td></td> </tr> <tr> <td>限定子</td> <td></td> </tr> <tr> <td colspan="2">«property» (定義: GML)</td> </tr> <tr> <td>sequenceNumber</td> <td>14</td> </tr> <tr> <td>inlineOrByReference</td> <td>inline</td> </tr> <tr> <td>isMetadata</td> <td>false</td> </tr> <tr> <td colspan="2">追加設定</td> </tr> </table>	多重度		多重度	0..1	Ordered	False	重複を許可	False	詳細		ステレオタイプ	GML::property	別名		可視性	Public	誘導可能性	誘導可能	集約	なし	スコープ	インスタンス	制約		限定子		«property» (定義: GML)		sequenceNumber	14	inlineOrByReference	inline	isMetadata	false	追加設定	
多重度																																																																									
多重度	0..*																																																																								
Ordered	False																																																																								
重複を許可	False																																																																								
詳細																																																																									
ステレオタイプ	GML::property																																																																								
別名																																																																									
可視性	Public																																																																								
誘導可能性	誘導可能																																																																								
集約	なし																																																																								
スコープ	インスタンス																																																																								
制約																																																																									
限定子																																																																									
«property» (定義: GML)																																																																									
sequenceNumber	13																																																																								
inlineOrByReference	inline																																																																								
isMetadata	false																																																																								
追加設定																																																																									
多重度																																																																									
多重度	0..1																																																																								
Ordered	False																																																																								
重複を許可	False																																																																								
詳細																																																																									
ステレオタイプ	GML::property																																																																								
別名																																																																									
可視性	Public																																																																								
誘導可能性	誘導可能																																																																								
集約	なし																																																																								
スコープ	インスタンス																																																																								
制約																																																																									
限定子																																																																									
«property» (定義: GML)																																																																									
sequenceNumber	14																																																																								
inlineOrByReference	inline																																																																								
isMetadata	false																																																																								
追加設定																																																																									

図 6-5 EA における「\_urbanfunction」の関連役割順番定義③

上記のとおり「XSD 出力」機能は、機能要件の目標を満たしている。

### 6-1-3. 検証結果 (XSD 整形)

#### 1) XSD 整形前後のファイルの比較確認

ShapeChange から出力した XSD を整形プログラムで整形し、「6-1-2. 検証結果 (XSD 変換)」で用いた「urbanfunction」を確認すると下図のとおりである。

```
<xs:complexType name="_UrbanFunctionType" abstract="true">
  <xs:complexContent>
    <xs:extension base="core:_CityObjectType">
      <xs:sequence>
        <xs:element name="class" minOccurs="0" type="gml:CodeType"/>
        <xs:element name="function" maxOccurs="unbounded" minOccurs="0" type="gml:CodeType"/>
        <xs:element name="usage" maxOccurs="unbounded" minOccurs="0" type="gml:CodeType"/>
        <xs:element name="validFrom" minOccurs="0" type="xs:date"/>
        <xs:element name="validFromType" minOccurs="0" type="gml:CodeType"/>
        <xs:element name="enactmentFiscalYear" minOccurs="0" type="xs:gYear"/>
        <xs:element name="validTo" minOccurs="0" type="xs:date"/>
        <xs:element name="validToType" minOccurs="0" type="gml:CodeType"/>
        <xs:element name="expirationFiscalYear" minOccurs="0" type="xs:gYear"/>
        <xs:element name="legalGrounds" minOccurs="0" type="xs:string"/>
        <xs:element name="custodian" minOccurs="0" type="xs:string"/>
        <xs:element name="notificationNumber" minOccurs="0" type="xs:string"/>
        <xs:element name="finalNotificationDate" minOccurs="0" type="xs:date"/>
        <xs:element name="finalNotificationNumber" minOccurs="0" type="xs:string"/>
        <xs:element name="urbanPlanType" minOccurs="0" type="gml:CodeType"/>
        <xs:element name="areaClassificationType" minOccurs="0" type="gml:CodeType"/>
        <xs:element name="nominalArea" minOccurs="0" type="gml:MeasureType"/>
        <xs:element name="prefecture" minOccurs="0" type="gml:CodeType"/>
        <xs:element name="city" minOccurs="0" type="gml:CodeType"/>
        <xs:element name="reference" minOccurs="0" type="xs:anyURI"/>
        <xs:element name="reason" minOccurs="0" type="gml:StringOrRefType"/>
        <xs:element name="note" minOccurs="0" type="gml:StringOrRefType"/>
        <xs:element name="surveyYear" minOccurs="0" type="xs:gYear"/>
        <xs:element name="table" maxOccurs="unbounded" minOccurs="0" type="urf:UrbanPlanningBasicSurveyTablePropertyType"/>
        <xs:element name="attributes" maxOccurs="unbounded" minOccurs="0" type="urf:UrbanPlanningBasicSurveyRecordPropertyType"/>
        <xs:element name="lod-2MultiSurface" minOccurs="0" type="gml:MultiSurfacePropertyType"/>
        <xs:element name="lod-1MultiSurface" minOccurs="0" type="gml:MultiSurfacePropertyType"/>
        <xs:element name="lod0MultiSurface" minOccurs="0" type="gml:MultiSurfacePropertyType"/>
        <xs:element name="lod-2MultiCurve" minOccurs="0" type="gml:MultiCurvePropertyType"/>
        <xs:element name="lod-1MultiCurve" minOccurs="0" type="gml:MultiCurvePropertyType"/>
        <xs:element name="lod0MultiCurve" minOccurs="0" type="gml:MultiCurvePropertyType"/>
        <xs:element name="lod-2MultiPoint" minOccurs="0" type="gml:MultiPointPropertyType"/>
        <xs:element name="lod-1MultiPoint" minOccurs="0" type="gml:MultiPointPropertyType"/>
        <xs:element name="lod0MultiPoint" minOccurs="0" type="gml:MultiPointPropertyType"/>
        <xs:element name="target" minOccurs="0" maxOccurs="unbounded" type="urf:TargetPropertyType"/>
        <xs:element name="keyValuePairAttribute" minOccurs="0" type="uro:KeyValuePairAttributePropertyType"/>
        <xs:element name="dataQualityAttribute" minOccurs="0" type="uro:DataQualityAttributePropertyType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

図 6-6 整形プログラムによる「urbanfunction」の出力

上図において、表 6-1 で示した No.1 (XS 接頭辞が設定される)、No.2 (REF に変わった Type をもとに戻す)、No.3 (プロパティは Name を先頭ににする) が修正されていることが確認できる。

さらに「target」の型が現行の XSD と同じ「TargetPropertyType」に修正されているが、これは No.4 (「TargetPropertyType」の Ref は「core:\_CityObject」とする) に伴う修正である。

※No.4 は下図のとおり修正されている

```
<complexType name="TargetPropertyType">
  <sequence minOccurs="0">
    <element ref="urf:Target"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

➡

```
<complexType name="TargetPropertyType">
  <xs:sequence minOccurs="0">
    <xs:element ref="core:_CityObject"/>
  </xs:sequence>
  <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

図 6-7 「TargetPropertyType」の Ref 修正

次に「AbstractConstructionType」の整形を確認すると、No.2 (REF に変わった Type をもとに戻す) に加え、No.5 (「core:\_CityObjectType」を継承するクラスの継承クラスは「core:AbstractCityObjectType」に置換される) と No.6 (「AbstractConstructionType」に役割「boundedBy」を挿入する) が修正されていることを確認できる。

```

<complexType abstract="true" name="AbstractConstructionType">
  <complexContent>
    <extension base="core:_CityObjectType">
      <sequence>
        <element minOccurs="0" name="conditionOfConstruction" type="uro:ConditionOfConstructionValueType"/>
        <element minOccurs="0" name="dateOfConstruction" type="xs:date"/>
        <element maxOccurs="unbounded" minOccurs="0" name="constructionEvent" type="uro:ConstructionEventPropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="elevation" type="uro:ElevationPropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="height" type="uro:HeightPropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="occupancy" type="uro:OccupancyPropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="consFacilityTypeAttribute" type="uro:FacilityTypeAttributePropertyType"/>
        <element minOccurs="0" name="consFacilityIdAttribute" type="uro:FacilityIdAttributePropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="consFacilityAttribute" type="uro:FacilityAttributePropertyType"/>
        <element minOccurs="0" name="consBaseAttribute" type="uro:ConstructionBaseAttributePropertyType"/>
        <element minOccurs="0" name="consStructureAttribute" type="uro:ConstructionStructureAttributePropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="consDisasterRiskAttribute" type="uro:DisasterRiskAttributePropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="consDisasterAttribute" type="uro:DisasterAttributePropertyType"/>
        <element maxOccurs="0" minOccurs="0" name="consDataQualityAttribute" type="uro:DataQualityAttributePropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="consKeyValuePairAttribute" type="uro:KeyValuePairAttributePropertyType"/>
        <element minOccurs="0" name="lod0Geometry" type="gml:GeometryPropertyType"/>
        <element minOccurs="0" name="lod1Geometry" type="gml:GeometryPropertyType"/>
        <element minOccurs="0" name="lod2Geometry" type="gml:GeometryPropertyType"/>
        <element minOccurs="0" name="lod3Geometry" type="gml:GeometryPropertyType"/>
        <element minOccurs="0" name="lod4Geometry" type="gml:GeometryPropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="constructionInstallation">
          <complexType>
            <sequence>
              <element ref="uro:ConstructionInstallation"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

<xs:complexType name="AbstractConstructionType" abstract="true">
  <xs:complexContent>
    <xs:extension base="core:AbstractCityObjectType">
      <xs:sequence>
        <xs:element name="conditionOfConstruction" minOccurs="0" type="uro:ConditionOfConstructionValueType"/>
        <xs:element name="dateOfConstruction" minOccurs="0" type="xs:date"/>
        <xs:element name="constructionEvent" maxOccurs="unbounded" minOccurs="0" type="uro:ConstructionEventPropertyType"/>
        <xs:element name="elevation" maxOccurs="unbounded" minOccurs="0" type="uro:ElevationPropertyType"/>
        <xs:element name="height" maxOccurs="unbounded" minOccurs="0" type="uro:HeightPropertyType"/>
        <xs:element name="occupancy" maxOccurs="unbounded" minOccurs="0" type="uro:OccupancyPropertyType"/>
        <xs:element name="consFacilityTypeAttribute" maxOccurs="unbounded" minOccurs="0" type="uro:FacilityTypeAttributePropertyType"/>
        <xs:element name="consFacilityIdAttribute" minOccurs="0" type="uro:FacilityIdAttributePropertyType"/>
        <xs:element name="consFacilityAttribute" maxOccurs="unbounded" minOccurs="0" type="uro:FacilityAttributePropertyType"/>
        <xs:element name="consBaseAttribute" minOccurs="0" type="uro:ConstructionBaseAttributePropertyType"/>
        <xs:element name="consStructureAttribute" minOccurs="0" type="uro:ConstructionStructureAttributePropertyType"/>
        <xs:element name="consDisasterRiskAttribute" maxOccurs="unbounded" minOccurs="0" type="uro:DisasterRiskAttributePropertyType"/>
        <xs:element name="consDisasterAttribute" maxOccurs="unbounded" minOccurs="0" type="uro:DisasterAttributePropertyType"/>
        <xs:element name="consDataQualityAttribute" maxOccurs="0" minOccurs="0" type="uro:DataQualityAttributePropertyType"/>
        <xs:element name="consKeyValuePairAttribute" maxOccurs="unbounded" minOccurs="0" type="uro:KeyValuePairAttributePropertyType"/>
        <xs:element name="lod0Geometry" minOccurs="0" type="gml:GeometryPropertyType"/>
        <xs:element name="lod1Geometry" minOccurs="0" type="gml:GeometryPropertyType"/>
        <xs:element name="lod2Geometry" minOccurs="0" type="gml:GeometryPropertyType"/>
        <xs:element name="lod3Geometry" minOccurs="0" type="gml:GeometryPropertyType"/>
        <xs:element name="lod4Geometry" minOccurs="0" type="gml:GeometryPropertyType"/>
        <xs:element name="constructionInstallation" maxOccurs="unbounded" minOccurs="0" type="uro:ConstructionInstallationPropertyType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

図 6-8 「AbstractConstructionType」の整形

また下図のとおり、No.7 (「bldg:\_AbstractBuildingType」を「bldg:AbstractBuildingType」に置換する) も対応されている。

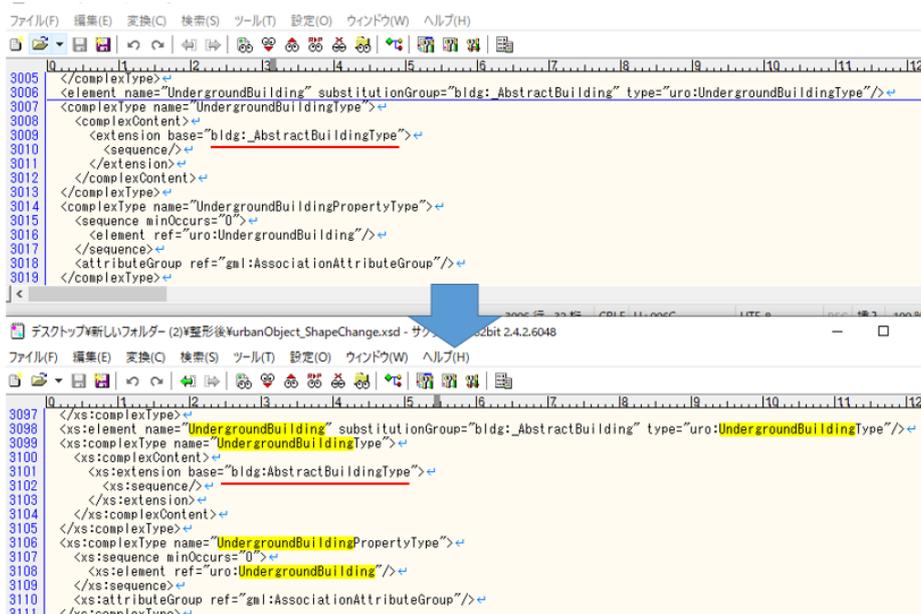


図 6-9 「bldg:AbstractBuildingType」の整形

なお下図のとおり、No.8（名称末尾に Type がついた列挙子の名称をもとに戻す）も対応されている。

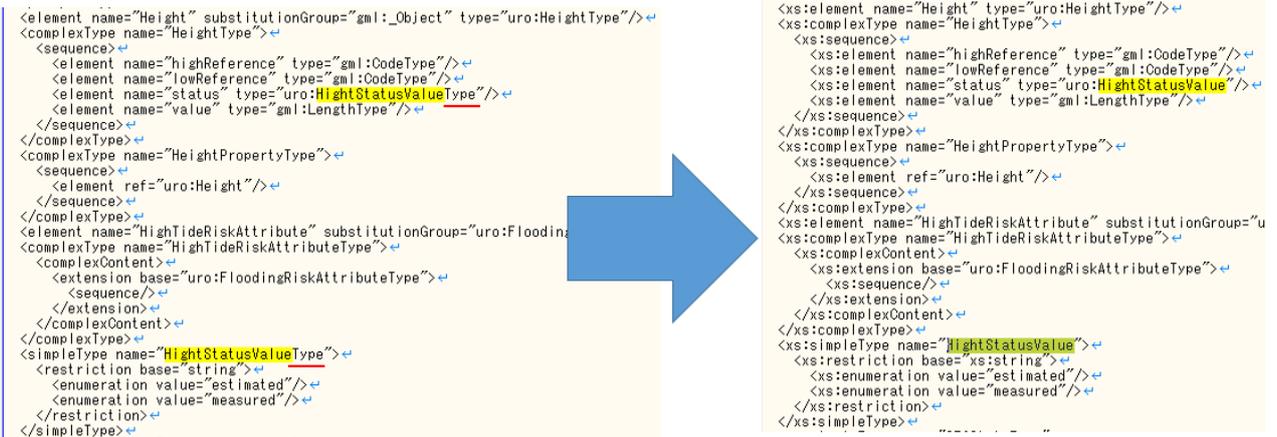


図 6-10 列挙子の名称修正

2) 現行 XSD と整形後 XSD の比較確認

No.9（現行 XSD と同内容のヘッダが記載される）については、コメント以外のヘッダ箇所のみを抽出し、タグとプロパティの順番のみ合わせて差分を確認した。

まず urf については、タグの閉じ方と、繰り返しをカットした以外に差分がないことが確認できる。

<pre> 1 &lt;?xml version="1.0" encoding="UTF-8"?&gt; 2 &lt;xs:schema 3   xmlns:xs="http://www.w3.org/2001/XMLSchema" 4   xmlns:core="http://www.opengis.net/cityml/2.0" 5   xmlns:gml="http://www.opengis.net/gml" 6   xmlns:urf="https://www.geospatial.jp/ur/urf/3.1" 7   xmlns:uro="https://www.geospatial.jp/ur/uro/3.1" 8   elementFormDefault="qualified" 9   attributeFormDefault="unqualified" 10  targetNamespace="https://www.geospatial.jp/ur/urf/3.1" 11  version="3.1.1"&gt; 12  &lt;xs:import namespace="http://www.opengis.net/gml" schemaLocation="http://schemas.opengis.net/ 13  &lt;xs:import namespace="http://www.opengis.net/cityml/2.0" schemaLocation="http://schemas.open 14  &lt;xs:import namespace="https://www.geospatial.jp/ur/uro/3.1" schemaLocation=".../uro/3.1/ur </pre>	<pre> 1 &lt;?xml version="1.0" encoding="UTF-8"?&gt; 2 &lt;xs:schema 3   xmlns:xs="http://www.w3.org/2001/XMLSchema" 4   xmlns:core="http://www.opengis.net/cityml/2.0" 5   xmlns:gml="http://www.opengis.net/gml" 6   xmlns:urf="https://www.geospatial.jp/ur/urf/3.1" 7   xmlns:uro="https://www.geospatial.jp/ur/uro/3.1" 8   elementFormDefault="qualified" 9   attributeFormDefault="unqualified" 10  targetNamespace="https://www.geospatial.jp/ur/urf/3.1" 11  version="3.1.1"&gt; 12  &lt;xs:import namespace="http://www.opengis.net/cityml/2.0" schemaLocation="http://schemas.open 13  &lt;xs:import namespace="http://www.opengis.net/gml" schemaLocation="http://schemas.opengis.net/ 14  &lt;xs:import namespace="https://www.geospatial.jp/ur/uro/3.1" schemaLocation=".../uro/3.1/ur </pre>
<pre> 11 12 &lt;/net/gml schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/&gt; 13 &lt;/net/cityml/2.0 schemaLocation="http://schemas.opengis.net/cityml/2.0/cityMLBase.xsd"/&gt; 14 &lt;/net/gml schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/&gt; 15 &lt;/gml:ur/urf/3.1 schemaLocation=".../uro/3.1/UrbanObject.xsd"/&gt;&lt;/xs:import&gt; </pre>	<pre> 11 12 &lt;/net/cityml/2.0 schemaLocation="http://schemas.opengis.net/cityml/2.0/cityMLBase.xsd"/&gt; 13 &lt;/net/gml schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/&gt; 14 &lt;/gml:ur/uro/3.1 schemaLocation=".../uro/3.1/UrbanObject.xsd"/&gt; </pre>

図 6-11 urf のヘッダ差分

同様に uro の差分を確認すると、こちらは差分がなかった。

<pre> 1 &lt;?xml version="1.0" encoding="UTF-8"?&gt; 2 &lt;xs:schema 3   xmlns:xs="http://www.w3.org/2001/XMLSchema" 4   xmlns:ids="http://www.opengis.net/cityml/building/2.0" 5   xmlns:core="http://www.opengis.net/cityml/2.0" 6   xmlns:furn="https://www.geospatial.jp/urf/furniture/2.0" 7   xmlns:gml="http://www.opengis.net/gml" 8   xmlns:land="http://www.opengis.net/cityml/landuse/2.0" 9   xmlns:trans="https://www.opengis.net/cityml/transportation/2.0" 10  xmlns:ur="https://www.geospatial.jp/ur/urf/3.1" 11  xmlns:uro="https://www.geospatial.jp/ur/uro/3.1" 12  xmlns:veg="http://www.opengis.net/cityml/vegetation/2.0" 13  xmlns:water="https://www.geospatial.jp/ur/water/2.0" 14  xmlns:group="https://www.geospatial.jp/ur/group/2.0" 15  xmlns:obj="https://www.geospatial.jp/ur/object/2.0" 16  elementFormDefault="qualified" attributeFormDefault="unqualified" 17  targetNamespace="https://www.geospatial.jp/ur/uro/3.1" 18  version="3.1.2"&gt; 19  &lt;xs:import namespace="http://www.opengis.net/ 20  &lt;xs:import namespace="http://www.opengis.net/ 21  &lt;xs:import namespace="http://www.opengis.net/cityml/building/2.0" schemaLocation="http://schemas.opengis.net/cityml/building/2.0/cityMLBuilding.xsd"/&gt; 22  &lt;xs:import namespace="http://www.opengis.net/cityml/2.0" schemaLocation="http://schemas.opengis.net/cityml/2.0/cityMLBase.xsd"/&gt; 23  &lt;xs:import namespace="https://www.geospatial.jp/urf/furniture/2.0" schemaLocation=".../urf/furniture/2.0/urfFurniture.xsd"/&gt; 24  &lt;xs:import namespace="http://www.opengis.net/cityml/landuse/2.0" schemaLocation="http://schemas.opengis.net/cityml/landuse/2.0/cityMLLanduse.xsd"/&gt; 25  &lt;xs:import namespace="http://www.opengis.net/cityml/transportation/2.0" schemaLocation="http://schemas.opengis.net/cityml/transportation/2.0/cityMLTransportation.xsd"/&gt; 26  &lt;xs:import namespace="https://www.geospatial.jp/ur/urf/3.1" schemaLocation=".../urf/3.1/UrbanObject.xsd"/&gt; 27  &lt;xs:import namespace="https://www.geospatial.jp/ur/uro/3.1" schemaLocation=".../uro/3.1/UrbanObject.xsd"/&gt; 28  &lt;xs:import namespace="http://www.opengis.net/cityml/vegetation/2.0" schemaLocation="http://schemas.opengis.net/cityml/vegetation/2.0/cityMLVegetation.xsd"/&gt; 29  &lt;xs:import namespace="https://www.geospatial.jp/ur/water/2.0" schemaLocation=".../ur/water/2.0/urWater.xsd"/&gt; 30  &lt;xs:import namespace="https://www.geospatial.jp/ur/group/2.0" schemaLocation=".../ur/group/2.0/urGroup.xsd"/&gt; 31  &lt;xs:import namespace="https://www.geospatial.jp/ur/object/2.0" schemaLocation=".../ur/object/2.0/urObject.xsd"/&gt; </pre>	<pre> 1 &lt;?xml version="1.0" encoding="UTF-8"?&gt; 2 &lt;xs:schema 3   xmlns:xs="http://www.w3.org/2001/XMLSchema" 4   xmlns:ids="http://www.opengis.net/cityml/building/2.0" 5   xmlns:core="http://www.opengis.net/cityml/2.0" 6   xmlns:furn="https://www.geospatial.jp/urf/furniture/2.0" 7   xmlns:gml="http://www.opengis.net/gml" 8   xmlns:land="http://www.opengis.net/cityml/landuse/2.0" 9   xmlns:trans="https://www.opengis.net/cityml/transportation/2.0" 10  xmlns:ur="https://www.geospatial.jp/ur/urf/3.1" 11  xmlns:uro="https://www.geospatial.jp/ur/uro/3.1" 12  xmlns:veg="http://www.opengis.net/cityml/vegetation/2.0" 13  xmlns:water="https://www.geospatial.jp/ur/water/2.0" 14  xmlns:group="https://www.geospatial.jp/ur/group/2.0" 15  xmlns:obj="https://www.geospatial.jp/ur/object/2.0" 16  elementFormDefault="qualified" attributeFormDefault="unqualified" 17  targetNamespace="https://www.geospatial.jp/ur/uro/3.1" 18  version="3.1.2"&gt; 19  &lt;xs:import namespace="http://www.opengis.net/ 20  &lt;xs:import namespace="http://www.opengis.net/ 21  &lt;xs:import namespace="http://www.opengis.net/cityml/building/2.0" schemaLocation="http://schemas.opengis.net/cityml/building/2.0/cityMLBuilding.xsd"/&gt; 22  &lt;xs:import namespace="http://www.opengis.net/cityml/2.0" schemaLocation="http://schemas.opengis.net/cityml/2.0/cityMLBase.xsd"/&gt; 23  &lt;xs:import namespace="https://www.geospatial.jp/urf/furniture/2.0" schemaLocation=".../urf/furniture/2.0/urfFurniture.xsd"/&gt; 24  &lt;xs:import namespace="http://www.opengis.net/cityml/landuse/2.0" schemaLocation="http://schemas.opengis.net/cityml/landuse/2.0/cityMLLanduse.xsd"/&gt; 25  &lt;xs:import namespace="http://www.opengis.net/cityml/transportation/2.0" schemaLocation="http://schemas.opengis.net/cityml/transportation/2.0/cityMLTransportation.xsd"/&gt; 26  &lt;xs:import namespace="https://www.geospatial.jp/ur/urf/3.1" schemaLocation=".../urf/3.1/UrbanObject.xsd"/&gt; 27  &lt;xs:import namespace="https://www.geospatial.jp/ur/uro/3.1" schemaLocation=".../uro/3.1/UrbanObject.xsd"/&gt; 28  &lt;xs:import namespace="http://www.opengis.net/cityml/vegetation/2.0" schemaLocation="http://schemas.opengis.net/cityml/vegetation/2.0/cityMLVegetation.xsd"/&gt; 29  &lt;xs:import namespace="https://www.geospatial.jp/ur/water/2.0" schemaLocation=".../ur/water/2.0/urWater.xsd"/&gt; 30  &lt;xs:import namespace="https://www.geospatial.jp/ur/group/2.0" schemaLocation=".../ur/group/2.0/urGroup.xsd"/&gt; 31  &lt;xs:import namespace="https://www.geospatial.jp/ur/object/2.0" schemaLocation=".../ur/object/2.0/urObject.xsd"/&gt; </pre>
---	---

図 6-12 uro のヘッダ差分

また下図のとおり、No.10 (「AbstractPointCloud」) の substitutionGroup は「gml:Feature」、継承は「gml:AbstractFeatureType」に設定する) も対応されている。

```

<element abstract="true" name="AbstractPointCloud" type="uro:AbstractPointCloudType"/>
<complexType abstract="true" name="AbstractPointCloudType">
  <complexContent>
    <sequence>
      <extension base="fixme:fixme"/>
    </sequence>
  </complexContent>
</complexType>
<complexType name="AbstractPointCloudPropertyType">
  <sequence minOccurs="0">
    <element ref="uro:AbstractPointCloud"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<xs:element name="AbstractPointCloud" abstract="true" substitutionGroup="gml:Feature" type="uro:AbstractPointCloudType"/>
<complexType name="AbstractPointCloudType" abstract="true">
  <complexContent>
    <xs:extension base="gml:AbstractFeatureType"/>
  </complexContent>
</complexType>
<complexType name="AbstractPointCloudPropertyType">
  <sequence minOccurs="0">
    <xs:element ref="uro:AbstractPointCloud"/>
  </sequence>
  <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

図 6-13 AbstractPointCloud の修正

3) 整形後 XSD の検索確認

No.11 (substitutionGroup="gml:Object"は出力されない) については、下図のとおり整形後の urf、uro の XSD で 1 行目を選択し、「substitutionGroup="gml:Object"」を下方検索してもヒットしないことが確認できる。

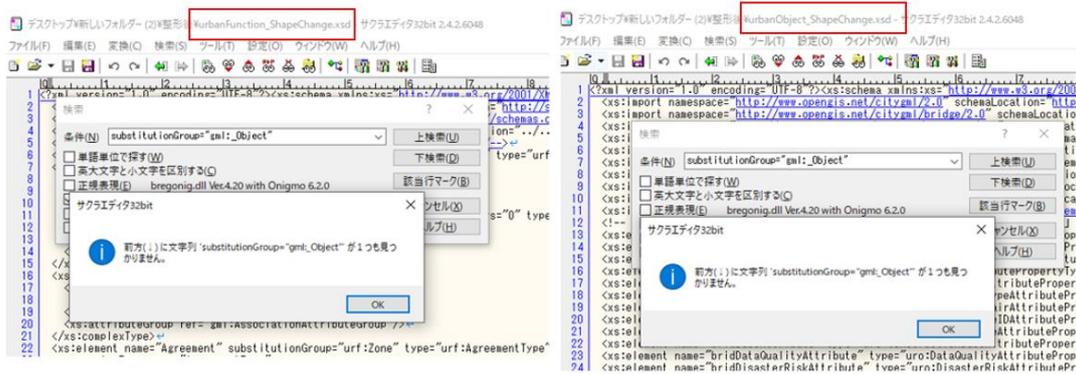


図 6-14 整形後 XSD の検索確認

#### 4) 整形後 XSD の追加行出力確認

最後に No.12 (ヘッダ部分の直後に追加対象クラスが追加される) については、uro において、整形プログラムで指定した追加行が挿入されていることを確認する。

追加する行を下図のとおり設定する。



図 6-15 整形プログラムによる追加行の設定

出力箇所であるヘッダの直後を確認すると、下図のとおり出力されている。

```

10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
11 <?xml version="1.0" encoding="UTF-8"?><xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:blgd="http://www.opengis.net/citygm/building/2.0" xmlns:brid="http://
12 <xs:import namespace="http://www.opengis.net/citygm/2.0" schemaLocation="http://schemas.opengis.net/citygm/2.0/cityGMBase.xsd"/>
13 <xs:import namespace="http://www.opengis.net/citygm/bridge/2.0" schemaLocation="http://schemas.opengis.net/citygm/bridge/2.0/bridge.xsd"/>
14 <xs:import namespace="http://www.opengis.net/citygm/building/2.0" schemaLocation="http://schemas.opengis.net/citygm/building/2.0/building.xsd"/>
15 <xs:import namespace="http://www.opengis.net/citygm/cityfurniture/2.0" schemaLocation="http://schemas.opengis.net/citygm/cityfurniture/2.0/cityfurniture.xsd"/>
16 <xs:import namespace="http://www.opengis.net/citygm/landuse/2.0" schemaLocation="http://schemas.opengis.net/citygm/landuse/2.0/landuse.xsd"/>
17 <xs:import namespace="http://www.opengis.net/citygm/transportation/2.0" schemaLocation="http://schemas.opengis.net/citygm/transportation/2.0/transportation.xsd"/>
18 <xs:import namespace="http://www.opengis.net/citygm/tunnel/2.0" schemaLocation="http://schemas.opengis.net/citygm/tunnel/2.0/tunnel.xsd"/>
19 <xs:import namespace="http://www.opengis.net/citygm/vegetation/2.0" schemaLocation="http://schemas.opengis.net/citygm/vegetation/2.0/vegetation.xsd"/>
20 <xs:import namespace="http://www.opengis.net/citygm/waterbody/2.0" schemaLocation="http://schemas.opengis.net/citygm/waterbody/2.0/waterbody.xsd"/>
21 <xs:import namespace="http://www.opengis.net/gml" schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
22 <!-- ===== 整形プログラムによる「_GenericApplicationPropertyOf」関係の追記 ↓ ===== -->
23 <xs:element name="blgdDataQualityAttribute" type="uro:DataQualityAttributePropertyType" substitutionGroup="blgd:_GenericApplicationPropertyOfAbstractBuilding"/>
24 <xs:element name="blgdDisasterRiskAttribute" type="uro:DisasterRiskAttributePropertyType" substitutionGroup="blgd:_GenericApplicationPropertyOfAbstractBuilding"/>
25 <xs:element name="blgdDmAttribute" type="uro:DmAttributePropertyType" substitutionGroup="blgd:_GenericApplicationPropertyOfAbstractBuilding"/>
26 <xs:element name="blgdFacilityAttribute" type="uro:FacilityAttributePropertyType" substitutionGroup="blgd:_GenericApplicationPropertyOfAbstractBuilding"/>
27 <xs:element name="blgdFacilityIdAttribute" type="uro:FacilityIdAttributePropertyType" substitutionGroup="blgd:_GenericApplicationPropertyOfAbstractBuilding"/>
28 <xs:element name="blgdFacilityTypeAttribute" type="uro:FacilityTypeAttributePropertyType" substitutionGroup="blgd:_GenericApplicationPropertyOfAbstractBuilding"/>
29 <xs:element name="blgdKeyValuePairAttribute" type="uro:KeyValuePairAttributePropertyType" substitutionGroup="blgd:_GenericApplicationPropertyOfAbstractBuilding"/>
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104 <xs:element name="waterBodyDetailAttribute" type="uro:WaterBodyDetailAttributePropertyType" substitutionGroup="wtr:_GenericApplicationPropertyOfWaterBody"/>
105 <xs:element name="wtrDataQualityAttribute" type="uro:DataQualityAttributePropertyType" substitutionGroup="wtr:_GenericApplicationPropertyOfWaterBody"/>
106 <xs:element name="wtrDmAttribute" type="uro:DmAttributePropertyType" substitutionGroup="wtr:_GenericApplicationPropertyOfWaterBody"/>
107 <xs:element name="wtrFacilityAttribute" type="uro:FacilityAttributePropertyType" substitutionGroup="wtr:_GenericApplicationPropertyOfWaterBody"/>
108 <xs:element name="wtrFacilityIdAttribute" type="uro:FacilityIdAttributePropertyType" substitutionGroup="wtr:_GenericApplicationPropertyOfWaterBody"/>
109 <xs:element name="wtrFacilityTypeAttribute" type="uro:FacilityTypeAttributePropertyType" substitutionGroup="wtr:_GenericApplicationPropertyOfWaterBody"/>
110 <xs:element name="wtrKeyValuePairAttribute" type="uro:KeyValuePairAttributePropertyType" substitutionGroup="wtr:_GenericApplicationPropertyOfWaterBody"/>
111 <!-- ===== 整形プログラムによる「_GenericApplicationPropertyOf」関係の追記 ↑ ===== -->
112 <!-- XML Schema document created by ShapeChange - http://shapechange.net/ -->
113 <xs:element name="AbstractConstruction" abstract="true" substitutionGroup="core:CityObject" type="uro:AbstractConstructionType"/>

```

図 6-16 整形プログラムによる追加行の出力

上記のとおり「XSD 整形」機能は、機能要件の目標を満たしている。

## 6-2. 概念モデルとドキュメント生成の検証

### 6-2-1. 検証方法

下記のとおり「ドキュメント生成」機能の機能要件を検証した。なお検証のカテゴリーは、EA と LutaML、スタイル、コンテンツに区分けする。

表 6-2 ドキュメント生成の機能要件検証

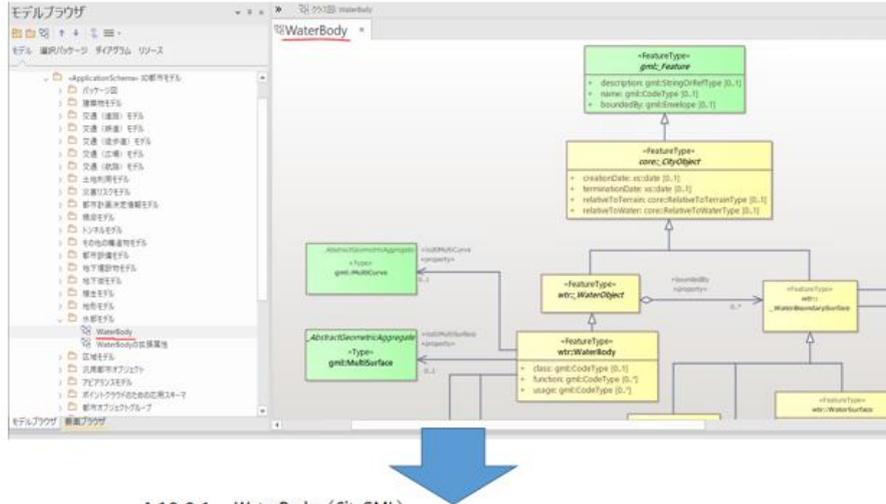
カテゴリー	No.	評価目標	検証方法
EA と LutaML	1	UML クラス図のファイルタイトルが EA の定義と相違なく表示される	出力ファイル UML クラス図において、EA の該当 UML クラス図名と相違ないことを確認する。
	2	自動レンダリングされる応用スキーマ文書で、EA で定義した説明文が出力され、定義もとのクラス情報が記載される	出力ファイルの応用スキーマ文書において、型、属性、役割に対し、EA で定義した説明文と、定義もとのクラス情報の表示、さらに関連役割の出力順を確認する。
	3	応用スキーマ文書における関連役割の出力順が、「3-2.開発内容」と相違なく出力される	
	4	コードリストが自動レンダリングされる	出力ファイルにおいてコードリストのレンダリングを確認する。
スタイル	1	表の中の図は自動採番されない	出力ファイルにおいて、表の中の図を確認する。
	2	OGC ModSpec が正常に表示されている	出力ファイルにおいて、OGC ModSpec の表示を確認する。
	3	「3-2.開発内容」で示した目次構成の変更が反映されている	出力ファイルにおいて、目次構成を確認する。
コンテンツ	1	標準製品仕様書と作業手順書がともに出力可能で、共有リソースへの修正が双方に同内容で反映される	標準製品仕様書と作業手順書がファイル出力され、共有ソースの修正が双方に反映されているのを確認する。

### 6-2-2. 検証結果 (EA と LutaML)

1) UML クラス図において、EA の該当 UML クラス図名と相違ないことを確認

全クラスを確認するのは困難であるため、ここでは水部で確認する。

まずクラス図名については、下図のとおり、EA のクラス図名と仕様書の UML クラス図名称が、相違なく表示されている。



4.19.2.1 WaterBody (CityGML)

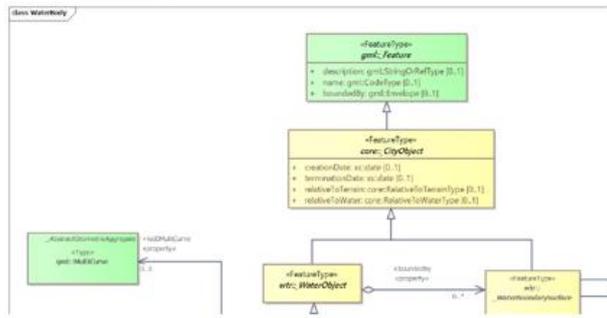


図 6-17 EA 出力した UML クラス図の名称

2) 応用スキーマ文書の確認

自動レンダリングされた応用スキーマ文書において、EA で定義した説明文が出力されることを確認する。まずクラスの型の説明が EA の定義どおり、EA の定義で示した画像ファイルを含めて出力されている。

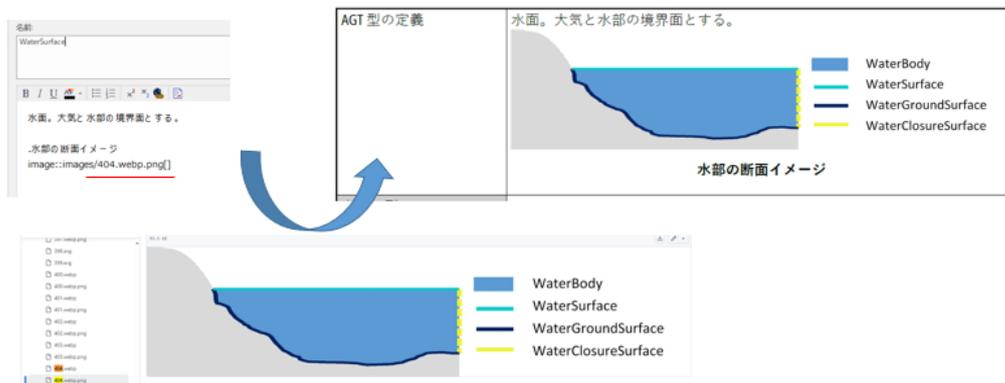


図 6-18 型の定義の説明

継承する属性の説明文も、下図のとおり定義もとのクラス情報とともに記載されている。



図 6-19 継承する属性の説明

同様に、自身に定義された属性の説明文も記載されている。

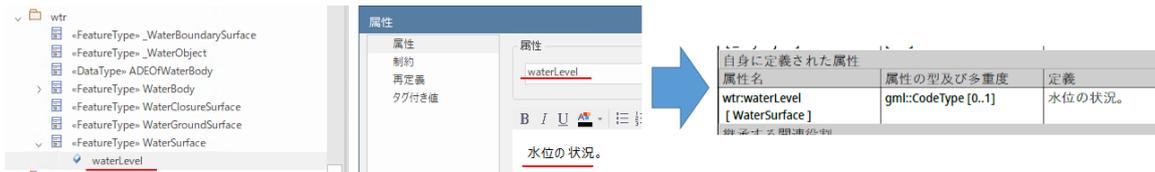


図 6-20 自身に定義された属性の説明

継承する関連役割の説明文も、下図のとおり定義もとのクラス情報とともに記載され、また出力順が「3-2. 開発内容」と相違ない。



図 6-21 継承する関連役割の説明

継承する関連役割		
関連役割名	関連役割の型及び多重度	定義
core:externalReference [_CityObject]	core:ExternalReference [0..*]	外部への参照。
gen:dateAttribute [_CityObject]	gen:dateAttribute [0..*]	日付型属性。属性を追加したい場合に使用する。
gen:doubleAttribute [_CityObject]	gen:doubleAttribute [0..*]	実数型属性。属性を追加したい場合に使用する。
gen:genericAttributeSet [_CityObject]	gen:genericAttributeSet [0..*]	汎用属性のセット。属性を追加したい場合に使用する。
gen:intAttribute [_CityObject]	gen:intAttribute [0..*]	整数型属性。属性を追加したい場合に使用する。
gen:measureAttribute [_CityObject]	gen:measureAttribute [0..*]	単位付き数値型属性。属性を追加したい場合に使用する。
gen:stringAttribute [_CityObject]	gen:stringAttribute [0..*]	文字列型属性。属性を追加したい場合に使用する。
gen:uriAttribute [_CityObject]	gen:uriAttribute [0..*]	URI型属性。属性を追加したい場合に使用する。
uro:pointCloud [_CityObject]	uro:AbstractPointCloud [0..1]	ポイントクラウドへの参照。
wtr:boundedBy [_WaterObject]	wtr:WaterBoundarySurface [0..*]	水部の外形を構成する水面 (wtr:WaterSurface) や水底面 (wtr:WaterGroundSurface) 等の境界面。

図 6-22 継承する関連役割の表示順

同様に、自身に定義された関連役割の説明文も記載され、また出力順が「3-2. 開発内容」と相違ない。

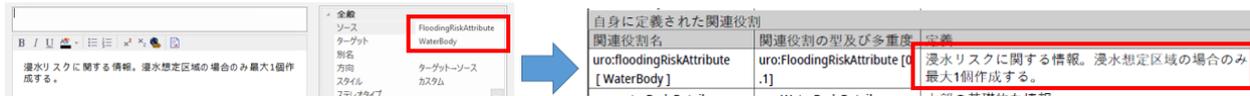


図 6-23 自身に定義された関連役割の説明

自身に定義された関連役割	関連役割名	関連役割の型及び多重度	定義
	uro:floodingRiskAttribute [WaterBody]	uro:FloodingRiskAttribute [0..1]	浸水リスクに関する情報。浸水想定区域の場合のみ最大1個作成する。
	uro:waterBodyDetailAttribute [WaterBody]	uro:WaterBodyDetailAttribute [0..1]	水部の基礎的な情報。
	uro:wtrDataQualityAttribute [WaterBody]	uro:DataQualityAttribute [0..1]	作成したデータの品質に関する情報。必須とする。公共測量成果又は基本測量成果となる場合はuro:PublicSurveyDataQualityAttributeを使用して記述する。公共測量成果又は基本測量成果とならない場合はuro:DataQualityAttributeを使用して記述する。
	uro:wtrDmAttribute [WaterBody]	uro:DmAttribute [0..*]	公共測量標準図式による図形表現に必要な情報。
	uro:wtrFacilityAttribute [WaterBody]	uro:FacilityAttribute [0..*]	uro:wtrFacilityTypeAttribute.classによって指定された分野における施設管理情報。
	uro:wtrFacilityIdAttribute [WaterBody]	uro:FacilityIdAttribute [0..1]	uro:wtrFacilityTypeAttribute.classによって指定された分野における施設の識別情報。
	uro:wtrFacilityTypeAttribute [WaterBody]	uro:FacilityTypeAttribute [0..*]	特定分野における施設の種類情報。
	uro:wtrKeyValuePairAttribute [WaterBody]	uro:KeyValuePairAttribute [0..*]	コード属性を拡張するための仕組み。コード値以外の属性を拡張する場合は、gen_GenericAttributeの下位型を使用する。
	wtr:lod0MultiCurve [WaterBody]	gml:MultiCurve [0..1]	LOD0の幾何形状。中心線とする。詳細は各モデルのLOD定義を参照すること。災害リスクモデルの浸水面を表現する場合には使用しない。
	wtr:lod1MultiSurface [WaterBody]	gml:MultiSurface [0..1]	LOD1の幾何形状。水面の範囲を面の集まりにより表現する。水平面に投影した場合に隣り合う水部のインスタンスは、連続でなければならない。各頂点の高さは、水位+標高となる。詳細は各モデルのLOD定義を参照すること。
	wtr:lod2Solid [WaterBody]	gml:_Solid [0..1]	LOD2の幾何形状。水面 (wtr:WaterSurface) と不規則三角網 (TIN) により表現する水底面 (wtr:WaterGroundSurface) を境界面とする水部の外形を示す立体により表現する。詳細は各モデルのLOD定義を参照すること。災害リスクモデルの浸水面を表現する場合には使用しない。
	wtr:lod3Solid [WaterBody]	gml:_Solid [0..1]	LOD3の幾何形状。水面 (wtr:WaterSurface) と不規則三角網 (TIN) を発生した水底面 (wtr:WaterGroundSurface) を境界面とする水部の外形を示す立体により表現する。詳細は各モデルのLOD定義を参照すること。災害リスクモデルの浸水面を表現する場合には使用しない。

図 6-24 自身に定義された関連役割の表示順

3) コードリストのレンダリングを確認

下図のとおり、LutaML が指定したコードリスト XML を用いて、コードリストの表を自動レンダリングしている。

```
[[toc4_19_04_01]]
==== WaterBody (CityGML)

==== WaterBody_class.xml

lutaml_gml_dictionary::iur/codelist/3.1/WaterBody_class.xml[template="gml_dict_template.liquid",context=dict]

4.19.4.1 WaterBody (CityGML)
4.19.4.1.1 WaterBody_class.xml
```

**表4-805**

ファイル名	WaterBody_class.xml
ファイル URL	https://www.geospatial.jp/iur/codlists/3.1/WaterBody_class.xml
コード	説明
1000	sea (海)
1010	tidal waterbody (潮汐水域)
1020	watercourse (水路)
1030	river / stream (河川/小川)
1060	lake / pond (湖)
1090	waterfall (滝)
1110	swamp (湿地、沼地)
1140	flooded land (浸水域)
1190	reservoir (貯水池)
9999	unknown (不明)

図 6-25 LuaML によるコードリストの自動レンダリング

上記のとおり EA と LutaML は、機能要件の目標を満たしている。

6-2-3. 検証結果 (スタイル)

1) 表の中の図を確認

出力ファイルにおいて確認する。

まず下図のとおり、表の中の図は自動採番されていない。

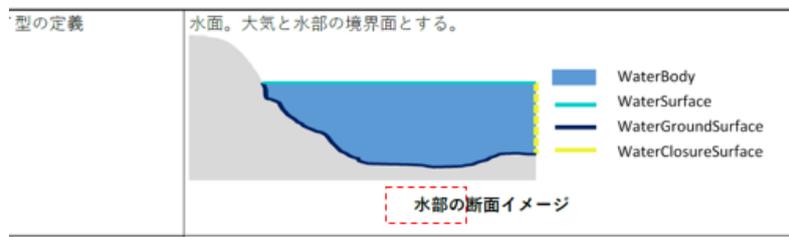


図 6-26 表の中の図は採番されない

2) OGC ModSpec の表示を確認

出力ファイルにおいて、OGC ModSpec の表示が、列名も日本語に変換された状態で確認できる。

表 D-1—要件分類 D-1: 妥当な交通（道路）オブジェクト

ID	/req/tran
対象の種類	3D都市モデル
説明	妥当な交通（道路）オブジェクト
規定	要件 D-1: 道路記述のLOD対応条件 要件 D-2: 道路3Dモデルの幾何オブジェクト要件 要件 D-3: tran:Roadの幾何座標一致条件 要件 D-4: LOD0 tran:Roadの交差禁止条件 要件 D-5: tran:Roadオブジェクトの区切り位置 要件 D-6: LOD2 tran:Roadの空間属性一致条件 要件 D-7: LOD3 tran:Roadの空間属性一致条件 要件 D-8: 交通領域の区分方法 要件 D-9: tran:TrafficAreaの重なり禁止条件 要件 D-10: 交通補助領域の区分方法 要件 D-11: tran:AuxiliaryTrafficAreaの重なり禁止条件

D.2 道路の記述とLOD

D.2.1 使用可能な地物型とLOD

CityGMLには、道路を記述するための地物型が用意されている。詳細度（LOD）ごとに、使用するべき地物型やその空間属性が異なる。

表 D-2—要件 D-1: 道路記述のLOD対応条件

ID	/req/tran/1
主題	3D都市モデル
分類	要件分類 D-1: 妥当な交通（道路）オブジェクト
明文	道路の記述には、指定されたLODに対応する地物型及び空間属性を使用する。

図 6-27 OGC ModSpec の表示

3) 目次構成を確認

出力ファイルから、標準製品仕様書と作業手順書の双方の「改訂履歴」が、「はじめに」の後にあることが確認できる。



図 6-28 標準製品仕様書の改訂履歴



図 6-29 作業手順書の改訂履歴

また作業手順書は 1 章始まりで、1 章に標準製品仕様書と同じ引用規格等がある。

<b>1</b>	<b>概要</b> .....	<b>1</b>
1.1	本書の目的.....	1
1.2	<b>引用規格等</b> .....	1
1.3	作業工程.....	2
1.4	本書で使用する用語.....	2
<b>2</b>	<b>製品仕様の決定</b> .....	<b>5</b>
2.1	概要.....	5
2.2	標準製品仕様書との比較.....	7

図 6-30 作業手順書の目次

上記のとおりスタイルは、機能要件の目標を満たしている。

## 6-2-4. 検証結果（コンテンツ）

### 1) ファイル出力と共有ソースの反映を確認

下図のとおり、共通リソースでテスト用の文言を記載して処理し、検証する。下記のとおりコンテンツは、機能要件の目標を満たしている。

```
[cols="2a,7a",options="header"]
.ファイル分割ルール
|===
| 基本となるファイル単位 | 分割ルール

| 第2次地域区画
|
| 緯線方向、経線方向に2等分に区切る「4分割」を基本とする。 +
| これはテストです。

.4分割の例
image:../common/images/con_001.webp.png[width="300"]

4分割したファイルであっても、ファイルサイズが上限を超える場合は、上限を超えるファイルのみを第3次地域区画に分割する。 +
```

図 6-31 共通リソースでテスト用の文言を記載

標準製品仕様書と作業手順書の該当箇所に、テスト用の文言が下記のとおり出力されている。

表7-3ーファイル分割ルール

基本となるファイル単位	分割ルール
第2次地域区画	緯線方向、経線方向に2等分に区切る「4分割」を基本とする。 <u>これはテストです。</u>

第2次地域区画

一辺約5km

4分割の例

4分割したファイルであっても、ファイルサイズが上限を超える場合は、上限を超える

図 6-32 標準製品仕様書のテスト修正出力

表6-19ーファイル分割ルール

基本となるファイル単位	分割ルール
第2次地域区画	緯線方向、経線方向に2等分に区切る「4分割」を基本とする。 <u>これはテストです。</u>

第2次地域区画

一辺約5km

4分割の例

4分割したファイルであっても、ファイルサイズが上限を超える場合は、上限を超える

図 6-33 作業手順書のテスト修正出力

## 7. 実証技術の非機能要件の品質検証

### 7-1. 概念モデルと符号化仕様の作成検証

#### 7-1-1. 検証方法

下記のとおり「XSD 整形」機能の非機能要件を検証した。

表 7-1 概念モデルと符号化仕様の非機能要件検証

機能	No.	評価目標	検証方法
XSD 整形	1	パラメータ設定とボタン押下のみで実行可能である	「4-3. 各機能の利用手順」で示したユーザインタフェースによる実行でエラー出力が発生しないまま終了メッセージが表示される。
	2	0.5 分以内で処理完了とする	実行開始と完了時の時間をデバッグ出力し確認する。

#### 7-1-2. 検証結果 (XSD 整形)

下図のとおり処理は正常完了し、処理時間も目標を満たしている。

	SC出力	整形後出力
urf	C:\Users\501829\Desktop\新しいフォルダー (2)\urbanFunction_ShapeChange.xsd	C:\Users\501829\Desktop\新しいフォルダー (2)\整形後\urbanFunction_ShapeChange.xsd
uro	C:\Users\501829\Desktop\新しいフォルダー (2)\urbanObject_ShapeChange.xsd	C:\Users\501829\Desktop\新しいフォルダー (2)\整形後\urbanObject_ShapeChange.xsd
URO-ver	3.2	

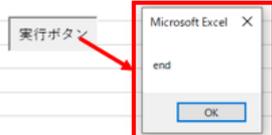
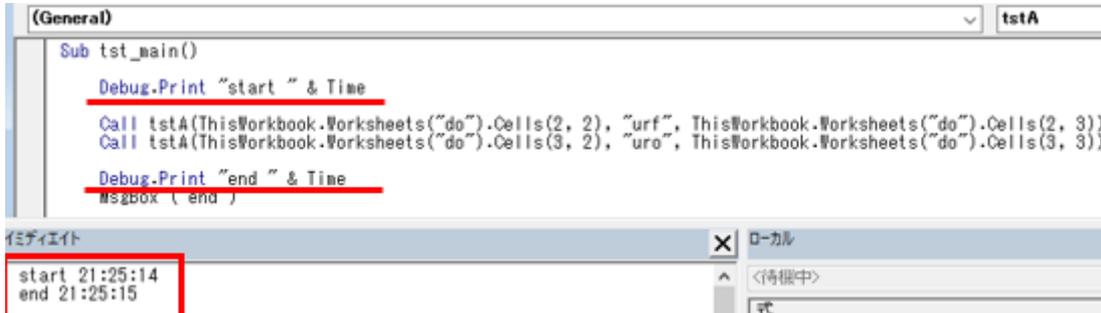


図 7-1 整形プログラムの実行



```

Sub tst_main()
    Debug.Print "start " & Time
    Call tstA(ThisWorkbook.Worksheets("do").Cells(2, 2), "urf", ThisWorkbook.Worksheets("do").Cells(2, 3))
    Call tstA(ThisWorkbook.Worksheets("do").Cells(3, 2), "uro", ThisWorkbook.Worksheets("do").Cells(3, 3))
    Debug.Print "end " & Time
    MsgBox ("end")
End Sub
    
```

ロケル

```

start 21:25:14
end 21:25:15
    
```

図 7-2 整形プログラムの処理時間

上記のとおり「XSD 整形」機能は、非機能要件の目標を満たしている。

## 7-2. 概念モデルとドキュメント生成の検証

### 7-2-1. 検証方法

下記のとおり「ドキュメント生成」機能の非機能要件を検証した。

表 7-2 概念モデルとドキュメント生成の非機能要件検証

機能	No.	評価目標	検証方法
ドキュメント生成	1	ファイル生成自体は、1 ファイルに対し 1 コマンドで実行可能であること	ファイル生成が 1 ファイルにつき 1 コマンドで実行されることを確認する。 ※カレントディレクトリ移動や、実行環境 Update のコマンドは含まない ※HTML のビューア用ラッパーについては考慮しない
	2	標準製品仕様書は 1～2 時間前後、作業手順書は 30 分前後で出力可能とする	実行開始と完了時の時間をコンソール出力し確認する。

上記要件を満たすため、管理者モードの PowerShell の対象ディレクトリで、実行環境 Update 後に、下記 3 つのコマンドを連続実行させ、ファイル生成と、コマンド実行前後の時間を確認する。

- 標準製品仕様書の生成
  - get-date
  - bundle exec metanorma sources/001-v5/document.adoc
  - get-date
- 作業手順書の生成
  - get-date
  - bundle exec metanorma sources/002-v5/document.adoc
  - get-date
- 標準製品仕様書と作業手順書の同時生成
  - get-date
  - bundle exec metanorma site generate --agree-to-terms
  - get-date

## 7-2-2. 検証結果（ドキュメント生成）

下図のとおり、1 コマンドで処理が流れ、処理時間も目標を満たしている。下記のとおり「ドキュメント生成」機能は、非機能要件の目標を満たしている。

```
PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau>
PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau> get-date
2025年1月26日 9:59:00

PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau> bundle exec metanorma sources/001-v5/document.adoc
[DEPRECATION] Template.register_tag is deprecated. Use Environment#register_tag instead. Called from C:/tools/ruby33/lib
~中略~
n="normative"/> from _アトラス化テキストをまとめる方法
java -Xss10m -Xmx3g -Djava.awt.headless=true -Duser.home=C:/Users/11704101 -jar C:/tools/ruby33/lib/ruby/gems/3.3.0/gems/mn2pdf-2.14/bin/mn2pdf.jar --xml-file "C:/Users/11704101/AppData/Local/Temp/document20250126-25184-2yzsqz.xml" --xsl-file "C:/tools/ruby33/lib/ruby/gems/3.3.0/gems/metanorma-plateau-0.1.15/lib/isodoc/plateau/plateau.international-standard.xsl" --pdf-file "C:/Users/11704101/Documents/GitHub/mn-samples-plateau/sources/002-v5/document.pdf" --param baseassetpath="C:/Users/11704101/Documents/GitHub/mn-samples-plateau/sources/002-v5" --syntax-highlight --font-manifest "C:/Users/11704101/AppData/Local/Temp/fontist_locations20250126-25184-u5if51.yml"
PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau> get-date
2025年1月26日 12:05:18

PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau>
```

図 7-3 標準製品仕様書の生成

```
PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau>
PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau> get-date
2025年1月25日 16:11:32

PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau> bundle exec metanorma sources/002-v5/document.adoc
[DEPRECATION] Template.register_tag is deprecated. Use Environment#register_tag instead. Called from C:/tools/ruby33/lib
~中略~
n="normative"/> from _アトラス化テキストをまとめる方法
java -Xss10m -Xmx3g -Djava.awt.headless=true -Duser.home=C:/Users/11704101 -jar C:/tools/ruby33/lib/ruby/gems/3.3.0/gems/mn2pdf-2.14/bin/mn2pdf.jar --xml-file "C:/Users/11704101/AppData/Local/Temp/document20250125-46476-r82j9a.xml" --xsl-file "C:/tools/ruby33/lib/ruby/gems/3.3.0/gems/metanorma-plateau-0.1.15/lib/isodoc/plateau/plateau.international-standard.xsl" --pdf-file "C:/Users/11704101/Documents/GitHub/mn-samples-plateau/sources/002-v5/document.pdf" --param baseassetpath="C:/Users/11704101/Documents/GitHub/mn-samples-plateau/sources/002-v5" --syntax-highlight --font-manifest "C:/Users/11704101/AppData/Local/Temp/fontist_locations20250125-46476-11c9s2.yml"
PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau> get-date
2025年1月25日 16:17:22

PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau>
```

図 7-4 作業手順書の生成

```
PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau>
PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau> get-date
2025年1月25日 20:26:16

PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau> bundle exec metanorma site generate --agree-to-terms
info! Compiling C:/Users/11704101/Documents/GitHub/mn-samples-plateau/sources/001-v5/document.adoc ...
~中略~
processor.rb:20:in `<top (required)>':
site has been generated at C:/Users/11704101/Documents/GitHub/mn-samples-plateau/_site
PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau> get-date
2025年1月25日 22:31:21

PS C:\Users\11704101\Documents\GitHub\mn-samples-plateau>
```

図 7-5 標準製品仕様書と作業手順書の同時生成

## 8. 成果と課題

7 章までのとおり、既存フローからの課題であった、ドキュメント生成の自動化と、ソースの一元化については達成できた。

本書の範囲外のため割愛したが、利便性向上や有用性向上についても EA からの情報を提供することで、プロジェクト横断的に情報の一元化を実現しており、XMI による情報のやり取りは今後も使えるシチュエーションが発生することが見込まれる。

他方、下記の課題も見えてきており、今後の対策を検討して、より可用性を向上させていくことを目指したいと考える。

表 8-1 実証実験で得られた課題

項目	課題	対応案
EA	<ul style="list-style-type: none"> <li>● 同じプロジェクトファイルをバージョン違いの EA で編集すると設定が書き換わる等の、ファイル共有しての作業でトラブルが生じた</li> </ul>	<ul style="list-style-type: none"> <li>● バージョンファイル形式の統一を図る</li> </ul>
ShapeChange	<ul style="list-style-type: none"> <li>● 当初想定よりも整形プログラムの対応範囲が大きくなってしまった</li> </ul>	<ul style="list-style-type: none"> <li>● EA や ShapeChange の Config 設定で修正内容をカバーできないか再検討する</li> </ul>
LutaML	<ul style="list-style-type: none"> <li>● 応用スキーマ文書において、該当クラスで利用しない行を明示させたい</li> <li>● EA から出力した UML クラス図の画像ファイルを、adoc 直書きで参照させているが、ファイル名が EA の ID であるため、管理が煩雑である</li> <li>● 列挙型の表は adoc の記載でレンダリングしている</li> <li>● 標準製品仕様書と作業手順書でコードリストのソースが別フォルダーになっている</li> </ul>	<ul style="list-style-type: none"> <li>● 何らかのフラグで明示させるように修正する</li> <li>● LutaML での自動配置を目指す</li> <li>● LutaML での自動生成を目指す</li> <li>● 共通化を検討する</li> </ul>
スタイル	<ul style="list-style-type: none"> <li>● 画像サイズ、表の列幅、改行等、スタイル調整が難しい</li> <li>● 改行後に一文字下げが行われる場合と行われない場合がある</li> <li>● 利用画像では PNG を利用しているが、SVG も利用したい（現状 SVG はサイズが制御できない）</li> </ul>	<ul style="list-style-type: none"> <li>● ノウハウの蓄積と共有を行う</li> <li>● スタイル修正を促す</li> <li>● 技術改善を促す</li> </ul>
コンテンツ	<ul style="list-style-type: none"> <li>● adoc と MetanormaADOC の相違に対する情報が不足しており、検証にファイル生成を繰り返さないといけなかったため、時間を要した</li> <li>● ソースとなる画像の縮尺がばらばらである</li> </ul>	<ul style="list-style-type: none"> <li>● ノウハウの蓄積と共有を行う</li> <li>● 画像の縮尺を統一する規則を検討する</li> <li>● 画像サイズの統一を検討する</li> </ul>

## 9. 参考資料

---

MDA については下記 OGC のレポートを参考にした。

<https://docs.ogc.org/dp/23-040.html>

EA、ShapeChange、Metanorma の利用に当たっては、下記の標準サイトを参考にした。

[https://www.sparxsystems.jp/products/EA/ea\\_zero.htm](https://www.sparxsystems.jp/products/EA/ea_zero.htm)

<https://www.sparxsystems.jp/help/>

<https://shapechange.net/>

<https://www.metanorma.org/>

ただし ShapeChange については標準サイトの更新が滞っているため、下記 GitHub も利用した。

<https://shapechange.github.io/ShapeChange/3.1.0/>

ShapeChange の Config については下記を参考にした。

<https://shapechange.net/get-started/config/>

<https://docs.ogc.org/per/21-035r1.html>

<https://github.com/ShapeChange/ShapeChange/blob/master/src/test/resources/cityGml/>

<https://github.com/opengeospatial/CityGML3.0-GML-Encoding/tree/main/resources/encoding>

Metanorma の開発サイト（GitHub）は下記である。

<https://github.com/metanorma>

<https://github.com/metanorma?q=LutaML&type=all&language=&sort=>

<https://github.com/metanorma?q=plateau&type=all&language=&sort=>

adoc については下記を参照にした。なお MetanormaADOC と基本的な文法は同じである。

<https://takumon.github.io/asciidoc-syntax-quick-reference-japanese-translation/>

<https://docs.asciidoctor.org/>

本書では割愛したが、HTML ビューラッパーについては下記を参照のこと。

<https://github.com/metanorma/firelight>

OGC ModSpec の仕様は下記を参照にした。

[https://portal.ogc.org/files/?artifact\\_id=34762](https://portal.ogc.org/files/?artifact_id=34762)

## 10. 用語

### A) アルファベット順

表 10-1 用語（アルファベット順）

No.	用語	説明
1	adoc	軽量マークアップ言語である AsciiDoc の略称又は AsciiDoc 形式で記述されたファイルの拡張子のこと。
2	bundle	複数の gem をまとめて取り扱う bundler ツールのコマンドである。 今回の技術調査では、Metanorma によるドキュメント自動生成のため、gem に対しインストールを行う「bundle install」、アップデートする「bundle update」、実行する「bundle exec」を用いた。
3	EA	Enterprise Architect の略称で、オーストラリアの SparxSystems 社が開発・販売している UML モデリングツールである。
4	EAPX	EA のプロジェクトファイルの旧型のファイル形式のこと、拡張子は eapx である。
5	gem	Ruby プログラムにおけるライブラリのパッケージのこと。
6	JAR	Java Archive の略称で、コンパイルされた複数の Java プログラムのリソースを一つにまとめ ZIP 形式で圧縮されたファイル及びそれを出力するツールのこと。 圧縮されたファイルの拡張子には「.jar」が使われ、対応する JRE 環境で JAR ファイルをキックすることで、プログラムを実行できる。
7	JRE	Java Runtime Environment の略称で、Java アプリケーションを実行するために必要なソフトウェアである。
8	JSON	JavaScript Object Notation の略称のことで、データ記述言語の 1 つである。 軽量のテキストベースのデータ交換用フォーマットとして用いられる。
9	liquid	Shopify 社による動的にページを作成する際に用いる言語又はその構文で記載されたファイルの拡張子のことで、Metanorma でもサポートされている。
10	LutaML	Ruby プログラムによるデータモデルに対するアクセスライブラリのこと。 Metanorma は LutaML を拡張コマンドとして用いることによって、UML 情報等をドキュメントに直接取り込み可能である。
11	MDA	Model Driven Architecture の略称で、モデル駆動型アーキテクチャのこと。 ソフトウェア設計・開発・実装手法の一種であり、実装方法に依存しないモデルから、実装となるプログラムを生成する仕組みである。 本技術調査では、MDA をドキュメント化のフローに導入することによって、モデリングツール等のソース情報から、具体的な実装となる PDF や HTML といった各種形式のドキュメントを、自動生成する仕組みを検証した。

No.	用語	説明
12	Metanorma	標準文書を作成・発行するため、主として Ruby で開発されたオープンソース・フレームワークであり、香港に拠点を置く Ribose 社がメンテナンスしている。 MetanormaADOC で記述されたコンテンツから、指定されたスタイル（テンプレート）で出力する。
13	MetanormaADOC	Metanorma AsciiDoc の略称で、AsciiDoc に一部拡張と制限を加えた言語である（ファイル拡張子は adoc である）。
14	OGC	Open Geospatial Consortium の略称で、1994 年に設立された国際的な非政府による標準化団体である。 世界中の 500 を超える商業、政府、非営利及び研究機関が協力しながら、地理空間のコンテンツやサービス、GIS データ処理、データ共有等に関する標準規格の開発と実装を行っている。
15	OGC ModSpec	OGC Modular Specification の略称で、要件や適合性を指定するための OG の仕様である（規格については、OGC 08-131r3 参照）。 本検証では要件、推奨、留意事項の定義で用いる。
16	QEA	EA のプロジェクトファイルの新型のファイル形式のこと、拡張子は qea である。
17	ShapeChange	XML 形式で記載される Config 設定に沿って、概念モデルを符号化仕様に変換するフリーツールである。 Java で開発されており、実行ファイルは JRE11 の環境で動く JAR ファイルである（本開発時点のバージョンは 3.1.0 である）。
18	XMI	XML Metadata Interchange の略称で、メタデータ情報を交換する標準規格であり、UML モデルの交換形式として利用されることが多い。
19	XML スキーマ	XML 文書の論理的構造を定義するスキーマのこと、XSD で記述される。
20	XSD	XML Schema Definition Language の略称で、XML スキーマを XML 形式で記述する言語のこと、XSD で記述したファイルの拡張子は「xsd」である。
21	YAML	YAML Ain't Markup Language の略称で、JSON と同様、構造化データを表現するデータ形式の一種であり、JSON と比べて人の目での可読性を高めるルールを強いられるのが特徴である。

B) 五十音順

表 10-2 用語集（五十音順）

No.	用語	説明
1	標準製品仕様書	本書における 3D 都市モデル標準製品仕様書の略称とする。3D 都市モデルには拡張製品仕様書が含まれるため、略称でも標準と明示する。
2	作業手順書	本書における 3D 都市モデル標準作業手順書の略称とする。

以上

モデル駆動型アーキテクチャによる  
3D 都市モデル標準ドキュメントの管理システム  
技術検証レポート

発行：2025年3月

委託者：国土交通省 都市局

受託者：アジア航測株式会社