Generating model-based standards: 19170, CityGML

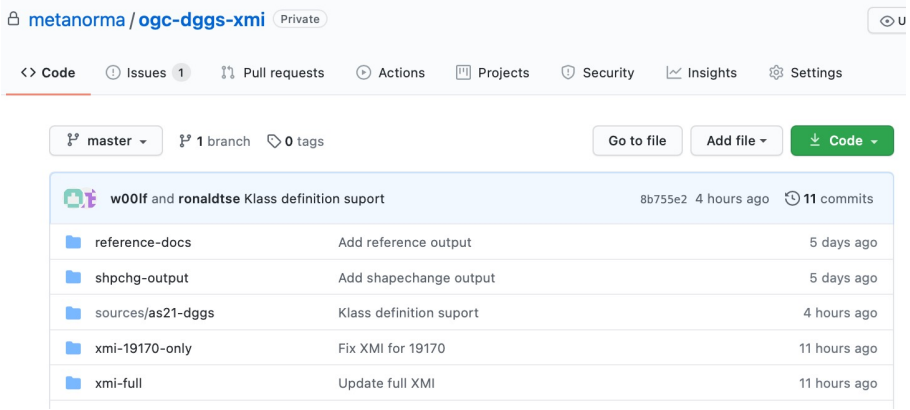
ISO/TC 211 Ad-hoc for Automated Documentation

May 20, 2021

Ronald Tse

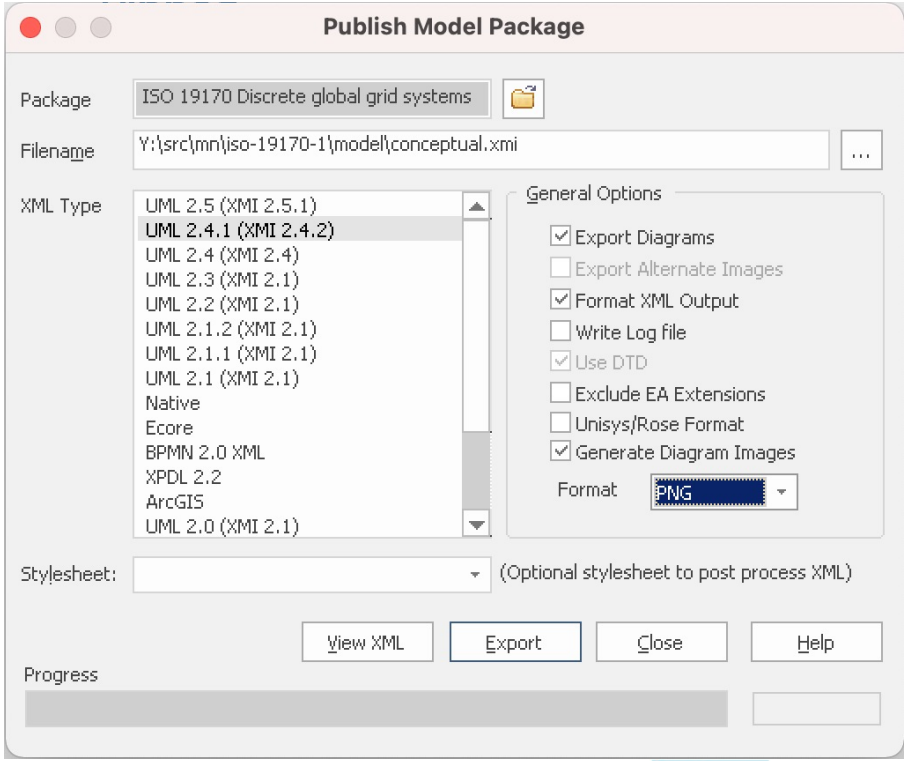
ISO 19170-1 in Metanorma

- Direct access to TC 211 harmonized model
- Copied over Metanorma document to new repository
- Previous version relied on Robert Gibb's program that extracted information from ShapeChange output
  - EA model => ShapeChange => HTML export => Custom program => Metanorma AsciiDoc
- Now:
  - EA model => Metanorma

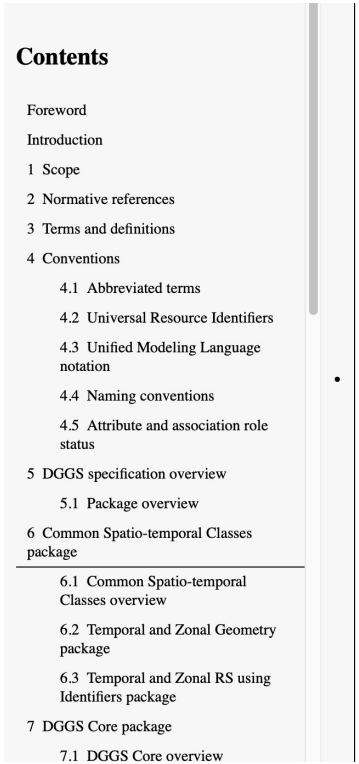


Exporting from EA to XMI 2.4.2

- Diagram images must be PNG for now…
  - EMF (the only vector choice) didn't work on macOS: Inkscape, ImageMagick, MS Word all reported as invalid image
  - (UPDATE: Knut says it works on Windows, need to find out how it works)



ISO 19170-1 fully generated via EA model (except missing packages)



Contents

ISO/TC 211 N 5025
Date: 2020-04-28
ISO/DIS 19170-1(E)
ISO/TC 211/WG 9
Secretariat: SN

Geographic information — Discrete Global Grid Systems Specifications —

Part 1:

Core Reference System and Operations, and Equal Area Earth Reference System

Information géographique — Système Global de Données Maillées Discrètes —

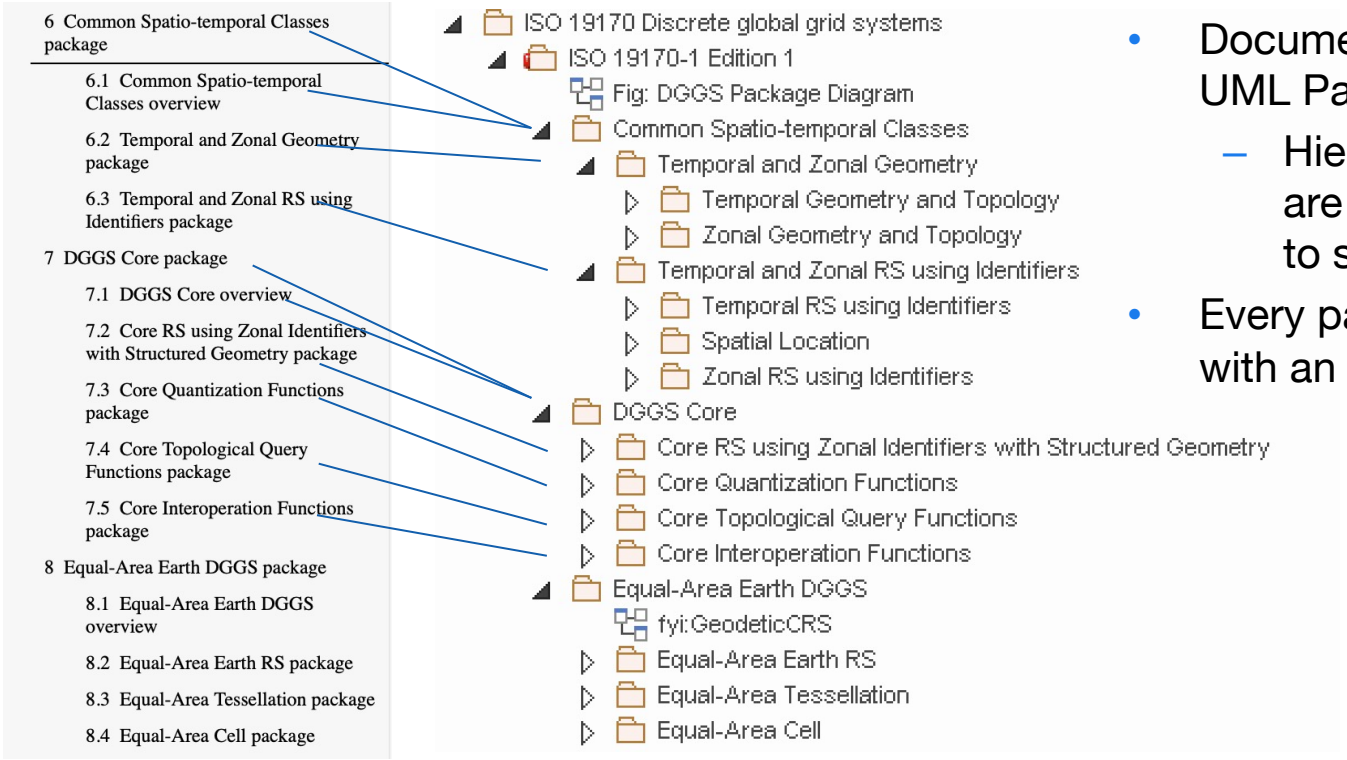
Partie 1:
Système de Référence et Opérations de Base, et Système de Référence Terrestre à Zone Égale

DIS stage

UML package representation structure

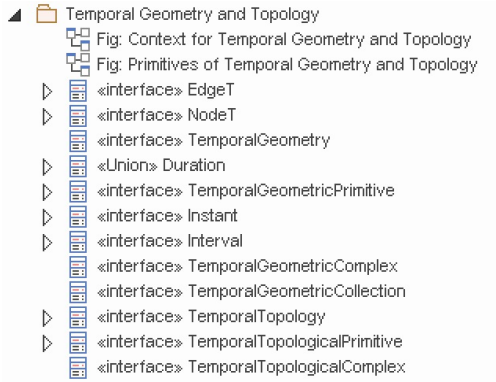
- ISO 19170-1 order:
  - Package
    - (diagrams)
    - Class
    - Enums
    - Data types
- CityGML order:
  - Class
    - (diagrams)
    - Class list
  - Data dictionary
  - …

ISO 19170: Document structure



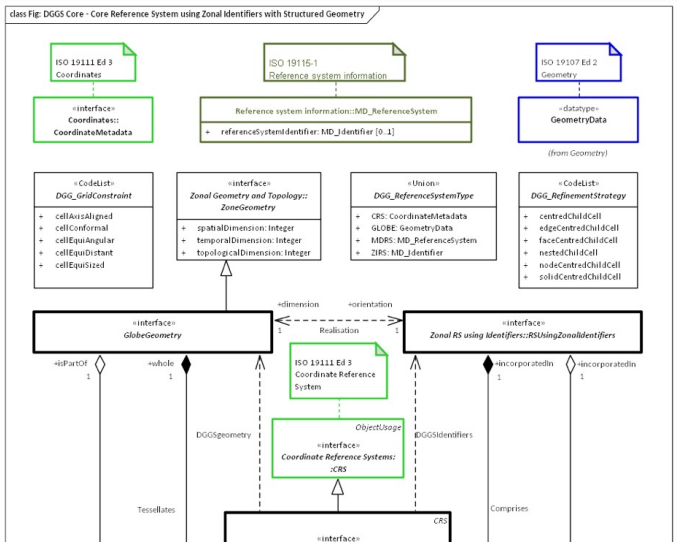
- Document structure follows UML Package structure
  - Hierarchical packages are rendered according to structure
- Every package clause starts with an “overview”

ISO 19170: Diagrams follow package structure



7.2 Core RS using Zonal Identifiers with Structured Geometry package

7.2.1 Core RS using Zonal Identifiers with Structured Geometry overview



ISO 19170: Representation of an UML class

Table 4 — Elements of Temporal Geometry and Topology::TemporalGeometry

Name:	TemporalGeometry			
Definition:	**Temporal Geometry** implements 1D **Geometry** on a **Temporal Coordinate System.**			
Stereotype:	interface			
Inheritance from:	ZoneSimpleGeometry, Geometry			
Generalization of:	TM_GeometricPrimitive, TemporalGeometricPrimitive, TemporalGeometricCollection			
Abstract:	true			
Associations:	*Association with:*	*Obligation*	*Maximum occurrence*	*Provides:*
	TemporalGeometry	C	*	element
	TemporalTopology	C	1	topology
Public attributes:	(none)			
Constraints:	`self.coordinateDimension ⇐ 1`			
	`self.rsid.CoordinateSystem = TemporalCS`			
	`self.spatialDimension.isEmpty = True`			
	`self.type.in{point,line} = True`			

ISO 19170: Enums

Table 59 — Elements of Core Topological Query Functions::RelativePosition

Name:	RelativePosition	
Definition:	Enumeration for the relative position of two geometries projected to a single uni-directional dimension, e.g. time. **NOTE** in this specification the relative position names follow those adopted by W3C and \<\<OGC16-071r2\>\>, which is more recent than \<\<ISO19108\>\>.	
Stereotype:	interface	
Abstract:		
Associations:	(none)	
Values:	*Name*	*Definition*
	Before	For this Period and another.Period: self.end \< another.Period.begin
	After	For this Period and another.Period: self.begin \> another.Period.end
	Meets	For this Period and another.Period: self.end = another.Period.begin

ISO 19170: Requirements external to the model

A.1 Common Spatio-temporal Classes package conformance categories

The requirement tests that apply to each conformance category in the Common Spatio-temporal Classes package are listed in Table A.1

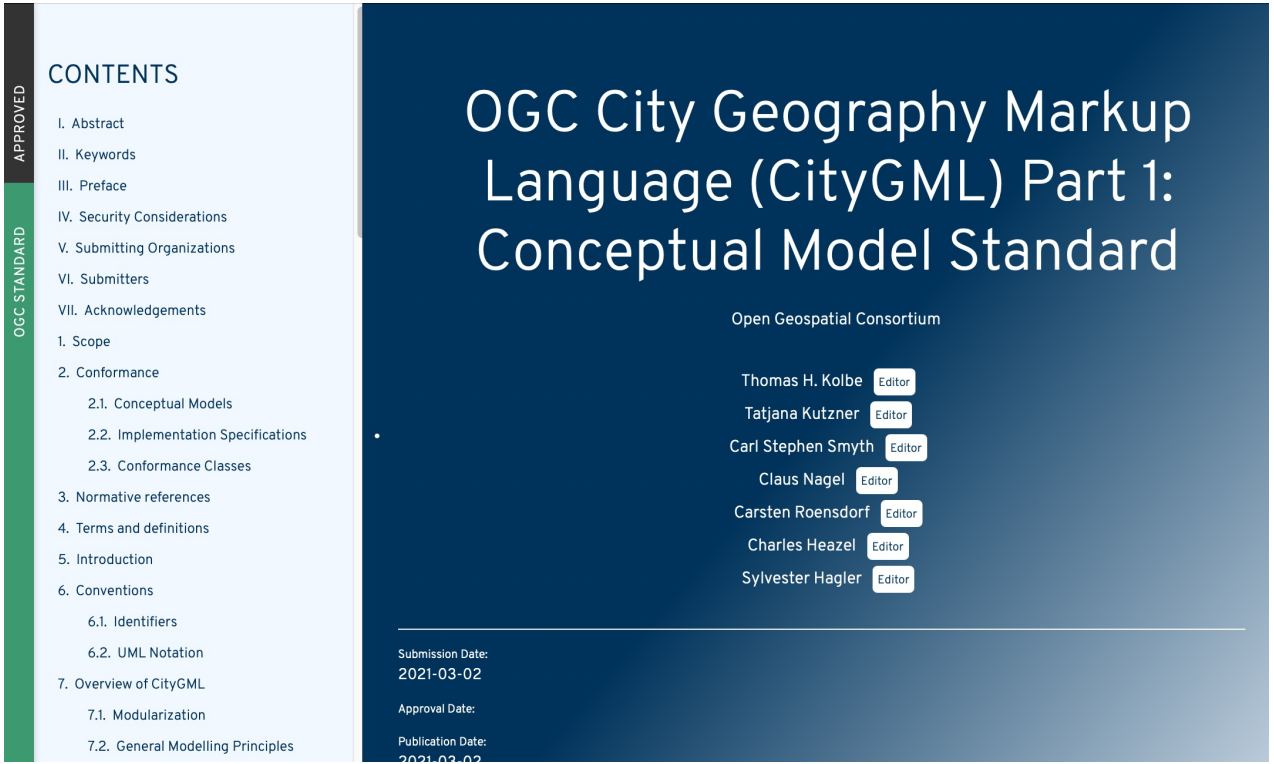
Table A.1 — Conformance classes for Common Spatio-temporal Classes package

Conformance class	Requirement tests
Temporal geometry and topology	Requirement A.1
Temporal reference systems using period identifiers	Requirement A.4
Spatial zone geometry and topology	Requirement A.2
Spatial reference systems using zonal identifiers	Requirement A.3, & Requirement A.5
Spatio-temporal zone geometry and topology	Requirement A.1, & Requirement A.2
Spatio-temporal reference systems using zonal identifiers	Requirement A.3, Requirement A.4, & Requirement A.5

Requirement A.1:
Common Spatio-temporal Classes — Temporal — Geometry

Abbreviation	conf/cc/temporal/geometry
Type	Basic
Requirement	6.2.2.2, requirement 1: http://www.opengis.net/spec/DGGS/2.0/req/cc/temporal/geometry
Reference subclause	6.2.2
Test purpose	To verify the common classes for temporal geometry and topology conform to the data model in Figure 3 and Figure 4, and defining tables in Table 5–Table 11.
Test method	Inspect documentation of the DGGS specification.

OGC CityGML 20-010



APPROVED

OGC STANDARD

CONTENTS

OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard

Open Geospatial Consortium

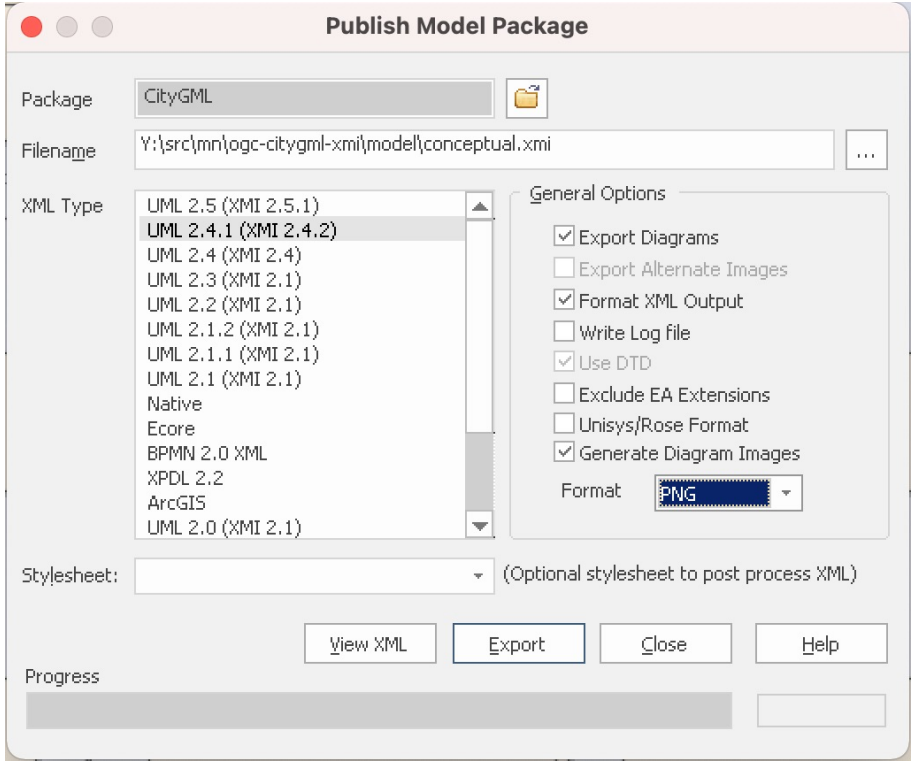
Thomas H. Kolbe Editor
Tatjana Kutzner Editor
Carl Stephen Smyth Editor
Claus Nagel Editor
Carsten Roensdorf Editor
Charles Heazel Editor
Sylvester Hagler Editor

Submission Date:
2021-03-02

Approval Date:

Publication Date:
2021-03-02

Same export options



CityGML: UML classes

10.1.4. Class GM_Object (ISO 19107: 2003)

Table 925

GM_Object

Definition:	GM_Object is the root class of the geometric object taxonomy and supports interfaces common to all geographically referenced geometric objects. GM_Object instances are sets of direct positions in a particular coordinate reference system. A GM_Object can be regarded as an infinite set of points that satisfies the set operation interfaces for a set of direct positions, TransfiniteSet<DirectPosition>. Since an infinite collection class cannot be implemented directly, a Boolean test for inclusion shall be provided by the GM_Object interface. This international standard concentrates on vector geometry classes, but future work may use GM_Object as a root class without modification. NOTE As a type, GM_Object does not have a well-defined default state or value representation as a data type. Instantiated subclasses of GM_Object will.
Subclass Of:	none
StereoType:	«type»
Constraint:	dimension() > boundary().dimension (Invariant):
Constraint:	boundary().notEmpty() implies boundary().dimension() = dimension() -1 (Invariant):
Constraint:	boundary().isEmpty() = isCycle() (Invariant):

Role name	Target class and multiplicity	Definition
	Geometry [1..1]	
	TransfiniteSet<DirectPosition> [1..1]	
	CV_DomainObject [1..1]	
CRS	CRS [0..1]	
CRS	SC_CRS [0..1]	

Note: Unless otherwise specified, all attribute and role names have the stereotype «Property»

Challenges: great power comes with great responsibilities

- Inaccurate annotations in model will carry on to document
  - Formatting errors in “Notes” field, such as “**Test Purpos**e”
  - Usage of RTF formatting mixed with AsciiDoc syntax (e.g. RTF formatting does not support definition lists)

Abbreviation 2.0/[req|conf]/core/rs/constraint/equal_area*Requirement::* _Equal Area?—?the DGGS Constraint of an Equal Area Earth Reference System SHALL be EqualArea_*Test Purpos*e:: To verify the DGGS ConstraintType of an Equal Area Earth Reference System is EqualArea.**Abbreviation**:: 2.0/[req|conf]/ea/ers/global_domain*Requirement::* _Global Domain?—?the DGGS Domain of an Equal Area Earth Reference System SHALL be the whole surface of the DGGS Reference Frame’s Earth Model_*Test Purpos*e:: To verify the DGGS Domain of an Equal Area Earth Reference System is the whole surface area of the DGGS Reference System’s Earth model.**Conformance Test Method**:: Inspect documentation of the DGGS specification.

- Diagrams cannot be exported as EMF (Windows Enhanced Metadata Format)
  - Enterprise Architect exported EMF files (Metafile format) can’t be read on macOS or Linux
  - https://github.com/metanorma/ogc-dggs-xmi/issues/17
  - Can’t be read by Word, Inkscape, LibreOffice
  - (UPDATE: Knut says it works on Windows, need to find out how it works)

Guidelines necessary for MBS

- Provide guidelines on input (model)
  - What to enter in “Notes” field?
    - What syntax?
  - What diagrams to make/display?
  - Requirements/Tests in/out of model?
  - Conventions for skipping hidden diagrams/models (e.g. named with “Spare”, “old: ”)

- Provide guidelines on output
  - What is the canonical format on output?
    - Per package => per diagram => per model?
    - MBS from Clause 4 or 5?
  - Should disallow insertion of arbitrary text/diagrams within MBS portion
  - Possible to have a TC 211-wide template for MBS generated text?

Thank you, questions welcome!