

# Open Geospatial Consortium

Submission Date: 2016-05-16

Approval Date: 2016-08-02

Publication Date: 2016-12-20

External identifier of this OGC® document: <http://www.opengis.net/doc/is/landinfra/1.0>

URL for this OGC® document: <http://docs.opengeospatial.org/is/15-111r1/15-111r1.html>

Internal reference number of this OGC® document: 15-111r1

Version: 1.0

Category: OGC® Conceptual Model Standard

Editor: Paul Scarponcini

Contributors:

Hans-Christoph Gruler (Survey), Erik Stubkjær (Land), Peter Axelsson, Lars Wikstrom (Rail)

## OGC® Land and Infrastructure Conceptual Model Standard (LandInfra)

### Copyright notice

Copyright © 2016 Open Geospatial Consortium

To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

### Warning

This document is an OGC Member approved international standard. This document is available on a royalty free, non-discriminatory basis. This formatted version is INFORMATIVE. The normative version is available at:

<http://docs.opengeospatial.org/is/15-111r1/15-111r1.html>

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OGC® Standard  
Document subtype: Conceptual Model  
Document stage: Approved  
Document language: English

## License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

## Contents

1. Scope.....	17
1.1 Relationship to Other Standards.....	17
2. Conformance.....	20
3. Normative References.....	21
4. Terms and Definitions.....	22
4.1 General Terms.....	22
4.2 Facility and Project Terms.....	26
4.3 Alignment Terms.....	29
4.4 Road Terms.....	30
4.5 Railway Terms.....	34
4.6 Survey Terms.....	35
4.7 Land Feature Terms.....	39
4.8 Land Division Terms.....	41
4.9 “Wet” Infrastructure Terms.....	44
5. Conventions.....	48
5.1 Abbreviations.....	48
5.2 UML Package and Class Diagrams.....	48
5.3 Requirements.....	49
6. Conceptual Modeling (informative).....	51
7. UML Conceptual Model (normative).....	53
7.1 Structural Overview of Requirements Classes.....	53
7.2 Core Requirements Class: LandInfra.....	58

7.2.1	LandInfra Requirements Class Classes.....	61
7.2.2	LandInfra Requirements Class Types.....	67
7.2.3	LandInfra Requirements Class Associated Classes .....	70
7.2.4	ISO 19103 Core Data Types.....	71
7.2.5	OGC Abstract Specification Topic 1 – Geometry Types with Extensions. ....	73
7.2.6	OGC Abstract Specification Topic 19 – Linear Referencing.....	75
7.2.7	ISO 19109 Application Schema.....	78
7.3	Requirements class: Facility.....	79
7.3.1	Facility Requirements Class Classes .....	81
7.3.2	Facility Requirements Class Associated Classes.....	87
7.3.3	Facility Part Subtypes .....	87
7.4	Requirements class: Project .....	88
7.4.1	Project Requirements Class Classes .....	89
7.4.2	Project Requirements Class Associated Classes.....	90
7.5	Requirements class: Alignment.....	91
7.5.1	Alignment .....	93
7.5.2	Alignment Set .....	95
7.5.3	Horizontal Alignment .....	95
7.5.4	Vertical Alignment.....	99
7.5.5	3D Alignment.....	103
7.5.6	Alignment for Linear Referencing.....	105
7.5.7	Alignment Summary .....	107
7.6	Requirements class: Road .....	108
7.6.1	Road .....	110
7.6.2	Road Element.....	111

7.6.3	Road Element Set.....	112
7.6.4	String Line .....	113
7.6.5	String Line Set .....	113
7.6.6	Surface .....	114
7.6.7	Surface Set .....	115
7.6.8	RoadCrossSection Requirements Class .....	115
7.7	Requirements Class: Railway.....	128
7.7.1	Railway .....	129
7.7.2	Alignment extensions.....	130
7.7.3	RailwayElement.....	131
7.7.4	Railway Element Set.....	132
7.7.5	Cant (Railway superelevation).....	133
7.8	Requirements class: Survey .....	137
7.8.1	Survey Requirements Class Classes .....	138
7.8.2	Requirements class: Observations .....	141
7.8.3	Observations Requirements Class Classes.....	144
7.8.4	Requirements class: Equipment.....	155
7.8.5	Equipment Requirements Class Classes .....	157
7.8.6	Requirements class: SurveyResults .....	168
7.8.7	SurveyResults Requirements Class Classes.....	170
7.8.8	OGC Abstract Specification Topic 20 – Observation and Measurements	177
7.9	Requirements class: LandFeature.....	179
7.9.1	Land Feature .....	181
7.9.2	Land Element.....	182
7.9.3	Land Surface .....	182

7.9.4	Land Layer .....	183
7.9.5	Solid Layer.....	183
7.9.6	Surfaces Layer .....	184
7.9.7	Linear Layer.....	185
7.9.8	TIN.....	188
7.10	Requirements class: LandDivision.....	194
7.10.1	LandDivision.....	197
7.10.2	InterestInLand.....	198
7.10.3	AdministrativeDivision.....	204
7.10.4	Statement.....	205
7.10.5	SurveyMonument.....	207
7.10.6	SpatialUnit .....	208
7.10.7	BoundingElements.....	210
7.11	Requirements class: Condominium.....	214
7.11.1	CondominiumUnit .....	216
7.11.2	CondominiumUseType.....	217
7.11.3	CondominiumBuilding .....	217
7.11.4	BuildingPart .....	218
7.11.5	BuildingPartType.....	218
7.11.6	CondominiumScheme.....	219
<b>Annex A</b>	<b>Conformance Class Abstract Test Suite (Normative) .....</b>	<b>220</b>
A.1	Core Conformance class: LandInfra .....	220
A.2	Conformance class: Facility.....	223
A.3	Conformance class: Project.....	224
A.4	Conformance class: Alignment .....	225

A.5	Conformance class: Road.....	227
A.6	Conformance class: RoadCrossSection.....	228
A.7	Conformance class: Railway.....	229
A.8	Conformance class: Survey.....	230
A.9	Conformance class: Observations.....	231
A.10	Conformance class: Equipment.....	232
A.11	Conformance class: SurveyResults.....	233
A.12	Conformance class: LandFeature.....	234
A.13	Conformance class: LandDivision.....	235
A.14	Conformance class: Condominium.....	236
<b>Annex B</b>	<b>Use Cases (informative).....</b>	<b>237</b>
B.1	Alignment / Road.....	237
B.1.1	Create, store, present Simple Horizontal Alignment.....	237
B.1.2	Create, modify, store, present Preliminary Design Alignment alternatives.....	239
B.1.3	Create, modify, store, present Detailed Design Alignment.....	241
B.1.4	Create, store, publish, utilize Construction Alignment.....	244
B.1.5	Create, store As-Built Alignment.....	246
B.1.6	Create, store, publish In-Service Alignment(s).....	247
B.2	Survey.....	248
B.2.1	S-1 Generate Control Points from Survey raw measurements.....	248
B.2.2	S-2 Interoperability between survey field systems and survey office systems.....	249
B.2.3	S-3 InfraGML is used as fieldbook reports.....	250
B.2.4	S-4 New sensor supports.....	251
B.2.5	S-5 Interoperability between survey field systems and monitoring systems.....	252

B.3	General .....	253
B.3.1	G-1 Porting old LandXML data to InfraGML .....	253
B.3.2	G-2 Porting InfraGML data back to LandXML.....	254
B.3.3	G-3 Digital preservation.....	255
B.4	Terrain .....	256
B.4.1	T-1 Store terrain data.....	256
B.5	Land Division.....	257
B.5.1	P-1 Right of way (ROW) for infrastructure project .....	257
B.5.2	P-2 Pointing out an existing real property boundary (Resurvey).....	259
B.5.3	P-3 Land parcel development /subdivision / change.....	261
B.5.4	P-4 Forming strata title / condominium .....	262
B.6	Railway.....	263
B.6.1	R-10 Handling chainage discontinuities between tracks.....	263
B.6.2	R-11 Conversion between representations of positions .....	264
B.6.3	R-20 Create, modify, store, present single objects along track.....	265
B.6.4	R-30 Create, modify, store and present multiple objects along track .....	266
B.6.5	R-40 Create, modify, store, present cant.....	267
<b>Annex C</b>	<b>Linear Referencing (informative)</b> .....	<b>268</b>
C.1	ISO 19148 - Linear referencing .....	268
C.2	Linear Referencing with Offsets .....	273
C.3	Alignments and Linear Referencing .....	275
<b>Annex D</b>	<b>Comparison with Other Standards (informative)</b> .....	<b>276</b>
D. 1	LandXML.....	276
D. 2	ISO 19152:2012 Land Administration Domain Model (LADM) .....	281
D.2.1	Party .....	281

D.2.2	Administrative units and ownership rights .....	281
D.2.3	Spatial units and 4 Surveying and spatial representations .....	283
D.2.4	Summary .....	283
D. 3	INSPIRE Data Specification .....	284
D.3.1	Theme Addresses .....	284
D.3.2	Theme Administrative Units.....	284
D.3.3	Theme Buildings.....	285
D.3.4	Cadastral Parcels.....	286
D.3.5	Theme Coordinate Reference Systems (CRS).....	287
D.3.6	Geographical Names.....	288
D. 4	The ePlan Model of the Intergovernmental Committee on Surveying and Mapping, Australia .....	289
Annex E	TIN Example .....	293
E. 1	Input Points .....	293
E. 2	Initial Triangles .....	294
E. 3	Constraints.....	295
E.3.1	Breakline.....	295
E.3.2	Soft Break.....	297
E.3.3	Control Contour.....	298
E.3.4	Break void .....	301
E.3.5	Drape void .....	304
E.3.6	Void .....	307
E.3.7	Hole .....	311
	Stop lines.....	311
	Maximum side length .....	311

Surface Boundary.....	311
<b>Annex F Revision history (informative).....</b>	<b>313</b>
<b>Annex G Bibliography .....</b>	<b>314</b>

## Figures

Figure 1. Requirements Classes as UML Packages with their dependencies.....	56
Figure 2. LandInfra Core Requirements Class .....	59
Figure 3. LandInfra Requirements Class Classes .....	61
Figure 4. LandInfra Requirements Class Types .....	67
Figure 5. ISO 19103 Core Data Types.....	71
Figure 6. OGC Abstract Specification Topic 1 - Geometry Types with Extensions.....	73
Figure 7. OGC Abstract Specification Topic 19 - Linear Referencing .....	75
Figure 8. OGC Abstract Specification Topic 19 – Linearly Located Events .....	76
Figure 9. ISO 19109 - Rules for application schema – Feature Type.....	78
Figure 10. Facility Requirements Class .....	80
Figure 11. Facility, FacilityPart, and FacilityPartRelationship Context Diagram.....	81
Figure 12. Element Context Diagram .....	85
Figure 13. Project Requirements Class.....	88
Figure 14. Project and ProjectPart Context Diagram.....	89
Figure 15. Alignment Requirements Class .....	92
Figure 16. Alignment Context Diagram .....	93
Figure 17. Horizontal Alignment.....	96
Figure 18. Vertical Alignment.....	100
Figure 19. 3D Alignment.....	103
Figure 20. Alignment as a Linear Element.....	105
Figure 21. Road Requirements Class.....	109
Figure 22. RoadElement context diagram .....	111
Figure 23. Road StringLine context diagram.....	113
Figure 24. Road Surface context diagram.....	114
Figure 25. RoadCrossSection Requirements Class.....	116
Figure 26. CrossSection context diagram .....	117
Figure 27. Two Lane CrossSection.....	118
Figure 28. CrossSection horizontal dimensioning (meters) and cross slopes.....	119
Figure 29. Pavement, base, and left gravel shoulder CrossSectionPoints and vertical dimensions (meters).....	119
Figure 30. Curb and gutter supplemental dimensions and CrossSectionPoints.....	120
Figure 31. Right side sidewalk, verge, and fill slope CrossSectionPoints and supplemental dimensions (meters).....	120
Figure 32. Left side ditch and cut slope CrossSectionPoints and supplemental dimensions (meters).....	121
Figure 33. CrossSectionComponent and CrossSectionPoint context diagram .....	121
Figure 34. CrossSectionArea context diagram .....	125
Figure 35. Two Lane Road with CUT and FILL CrossSectionAreas with additional CrossSectionPoints .....	127
Figure 36 - Railway Requirements class .....	129

Figure 37. RailwayElement context diagram.....	131
Figure 38 - Cant context diagram .....	134
Figure 39 - Cross section of left-sided CantEvents .....	135
Figure 40. Survey Requirements Class.....	138
Figure 41. Survey Class .....	139
Figure 42. SurveyObservation and OGC Abstract Specification Topic20 .....	141
Figure 43. Observations Requirements Class .....	143
Figure 44. Setup, InstrumentPoint, PanoramaImage and SetupParameters Context Diagram.....	144
Figure 45. InstrumentPoint and Targetpoint.....	146
Figure 46. SurveyObservation Classes with TargetPoint results.....	147
Figure 47. SurveyObservation Classes with no TargetPoint results.....	154
Figure 48. Equipment Requirements Class.....	156
Figure 49. Equipment, SurveySensor and Calibration Context Diagram.....	157
Figure 50. SurveySensor derived Classes.....	160
Figure 51. ObservationCorrection Classes.....	164
Figure 52. SurveyResults Requirements Class .....	169
Figure 53. SurveyResults and derived Classes .....	171
Figure 54. OGC Abstract Specification Topic 20 – Sampling Feature .....	177
Figure 55. LandFeature Requirements Class.....	180
Figure 56. Land Features .....	181
Figure 57. LandLayer context diagram.....	183
Figure 58. LinearLayer context diagram .....	185
Figure 59. TIN context diagram.....	189
Figure 60. LandDivision Requirements Class .....	196
Figure 61. LandDivision context diagram .....	197
Figure 62. InterestInLand context diagram.....	198
Figure 63. An Easement (gray) across four LandParcels .....	203
Figure 64. AdministrativeDivision Context Diagram.....	204
Figure 65. Statement and SurveyMonument context diagram.....	205
Figure 66. SpatialUnit context diagram .....	209
Figure 67. BoundingElement context diagram .....	210
Figure 68. Condominium Requirements Class Classes .....	215
Figure 69. Condominium context diagram .....	216

## Tables

Table 1. CrossSectionComponents .....	123
Table 2. CrossSectionComponent CrossSectionPoints.....	125
Table 3. LandInfra / LandXML corresponding concepts .....	280
Table 4. LandInfra / LADM corresponding concepts.....	283

## **i. Abstract**

This OGC Land and Infrastructure Conceptual Model Standard presents the implementation-independent concepts supporting land and civil engineering infrastructure facilities. Conceptual model subject areas include facilities, projects, alignment, road, rail, survey, land features, land division, and wet infrastructure (storm drainage, wastewater, and water distribution systems). The initial release of this standard includes all of these subject areas except wet infrastructure, which is anticipated to be released as a future extension.

This standard assumes the reader has a basic understanding of surveying and civil engineering concepts.

## **ii. Keywords**

The following are keywords to be used by search engines and document catalogues.

OGC document, LandInfra, Infra, infrastructure, civil, survey, land parcel, land feature, terrain, road, rail, alignment

## **iii. Preface**

After reviewing the existing LandXML format [1], the participants involved decided [2] that a new standard based on a subset of LandXML functionality was desired. Further, this new work activity would be use case driven, consistent with the OGC standards baseline, implemented with GML, and supported by a UML conceptual model. Called InfraGML, this new standard would:

- be supported by a recognized, dependable Standards Developing Organization, OGC
- align with existing OGC (and ISO TC211 and SQL/MM) standards, including the modular spec
- benefit from functionality already supported by GML, including features, geometry, coordinate reference systems, linear referencing, and surface modeling (TIN)
- initially focus on alignments/roads, survey, and land parcels, the subject areas which have identified need and committed resources for development
- using modular extensions, be able to expand into other areas (e.g., “wet” infrastructure pipe networks) as resources become available
- be use-case driven
- be based on a UML conceptual model developed prior to GML (and any other future) encoding
- have more up-to-date functionality

- be synchronized with the concurrent efforts by buildingSMART International in their development of Infrastructure-based Industry Foundation Classes (IFCs)
- be more easily integrated with CityGML and TransXML

The following steps for the new standards activity included:

1. **Subject area identification:** The list of initial subject areas for InfraGML were identified. Additionally, other areas may be identified as possible future extensions.
2. **Use case definition:** Use cases relevant to each subject area were identified and described. This was harmonized with the bSI Alignment Project use cases, the building SMART Alliance (bSA) BIM-GIS use case list, and any forthcoming use case definitions from the ISO TC211 GIS-BIM ad hoc committee.
3. **Conceptual modeling:** A UML Conceptual Model of the initial subject areas was developed, including defining Core functionality. This model was harmonized with the model developed by the bSI Alignment Project.
4. **RFC:** Prior to any GML encoding, a public Request for Comment was issued based upon the UML conceptual model [3]. This RFC requested input from the existing LandXML community (and others), including users as well as software developers. The conceptual model would then be revised accordingly.
5. **GML encoding:** A GML 3.2.1 / 3.3 compatible standard was then to be developed based on the revised UML conceptual model.
6. **Extensions:** Similar steps would be followed for subsequent subject areas.

This OGC document provides the revised UML Conceptual Model from step 4 above. In accordance with the use cases, this model standardizes a single set of consistent, implementation-independent concepts for the identified subject areas. The subject areas can be implemented in any potential implementation-specific standards such as InfraGML. The Land and Infrastructure Domain Working Group (DWG) and Standards Working Group (SWG) agreed that the implementation-independent Conceptual Model should first be adopted by the OGC. To ensure harmony among other OGC Standards and across other possible Infra implementations the approval should be prior to the creation of the InfraGML implementation standard.

The alignment subject area use cases and conceptual model were developed jointly with the buildingSMART International (bSI) Infrastructure Room IfcAlignment project [4] team to insure compatibility between the anticipated GML and IFC standards. Joint work continues to ensure compatibility with future bSI infrastructure projects such as IfcRoad.

Future extensions to this Land and Infrastructure Conceptual Model Standard that will address wet infrastructure are anticipated.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

*Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.*

**iv. Submitting organizations**

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

Bentley Systems, Inc.

Leica Geosystems

Aalborg University, Dept. of Development & Planning

Swedish Transport Administration

Trimble

Vianova Systems AS

Autodesk

**v. Submitters**

All questions regarding this submission should be directed to the editor or the submitters:

Name	Affiliation
Paul Scarponcini, SWG chair	Bentley Systems, Inc.
Hans-Christoph Gruler, SWG co-chair	Leica Geosystems
Erik Stubkjær	Aalborg University, Dept. of Development & Planning
Peter Axelsson	Swedish Transport Administration
Leif Granholm	Trimble
Johnny Jensen	Vianova Systems AS
Thomas Liebich	buildingSMART International
Orest Halustchak	Autodesk

## 1. Scope

The scope of the Land and Infrastructure Conceptual Model is land and civil engineering infrastructure facilities. Anticipated subject areas include facilities, projects, alignment, road, railway, survey, land features, land division, and “wet” infrastructure (storm drainage, wastewater, and water distribution systems). The initial release of this standard is targeted to support all of these except wet infrastructure.

Land provides the environment upon which infrastructure facilities exist. This standard includes division of land based on administrative (jurisdictions and districts) as well as interests in land (e.g., land parcels, easements, and condominiums). The standard also includes support for topography (terrain) as well as subsurface information. Finally, this standard regards the surveying needed to locate infrastructure facilities on the terrain in compliance with interests in land.

Infrastructure facilities are improvements constructed and operating on land. This standard includes support for information about civil engineered facilities, such as roads and railway and in the future, “wet” infrastructure including storm drainage, wastewater, and water distribution systems. Though often not considered to be infrastructure, buildings are included to a limited extent. The contributors to this standard do not anticipate that the design of utilities, such as electrical distribution, gas networks, and telecommunications will be explicitly included as facilities, though the detailed location of such facilities may at some point be added for referenceability. However, the rough location and legal protection of such facilities are covered by this standard; see SuperficieObject and Easement, subsections of 7.10.2, InterestsInLand.

This OGC<sup>®</sup> standard identifies concepts relevant to the identified subject areas, based upon a set of defined use cases. It is anticipated that future OGC infrastructure standards will implement these concepts in various implementation languages, such as GML.

### 1.1 Relationship to Other Standards

LandXML: This LandInfra Standard extends a subset of what is contained within LandXML 1.2 [1]. As a Conceptual Model Standard, this OGC standard defines some of the concepts which were implemented in LandXML 1.2. For a more detailed comparison, see Annex D. 1.

This standard is related to and/or consistent with the following OGC and ISO standards.

OGC Abstract Specification Topic 1 – Feature Geometry: This LandInfra standard relies heavily upon the geometry types defined in this Abstract Specification for spatial representations. LandInfra adds some additional geometry types not included in Topic 1.

OGC Abstract Specification Topic 2 – Spatial Referencing by Coordinates: This LandInfra Standard relies upon Topic 2 for the definition of coordinate reference systems.

OGC Abstract Specification Topic 19 – Linear Referencing: This LandInfra Standard relies heavily upon the linear referencing types defined in this Abstract Specification for linearly referenced locations.

OGC Abstract Specification Topic 20 – Observations and Measurements: This LandInfra Standard relies upon the observation acts, their results and sampling features defined in this Abstract Specification.

ISO 19130, Imagery sensor models for geopositioning: This LandInfra Standard relies upon the sensor data parameters for imagery sensors defined in this ISO TC211 standard.

ISO 19115, Metadata: This LandInfra Standard relies upon the metadata describing geospatial imagery defined in this ISO TC211 standard.

CityGML [5]: There are some potential overlaps between this LandInfra Standard and CityGML. Many of the FeatureTypes in LandInfra appear as CityGML CityObject subclasses such as Building (Building), Road (Road), Railway (Rail), LandSurface (ReliefFeature), LandFeature (WaterBody, VegetationObject), and AdministrativeDivision (LandUse). The CityGML feature name is shown in parentheses. However, the level of detail attached to these features is significantly different: CityGML focuses on Buildings whereas LandInfra focuses more on land and infrastructure facilities other than buildings.

IndoorGML [6]: This standard regards building and building parts used for access. Identification of these parts may be supplemented with indoor connectivity graphs and used for indoor navigation in accordance with IndoorGML. Analogous extensions to outdoor is anticipated.

ISO 19152, LADM [7]: This LandInfra Standard addresses a subset of LADM. Consistency requests motivate deviations from LADM, but compares to LADM classes.

ISO 19103, Conceptual Schema Language: This LandInfra Standard relies heavily upon the base data types defined in this ISO TC211 standard.

ISO 19109, Rules for Application Schema: This LandInfra Standard supports the General Feature Model as defined in this ISO TC211 standard.

bSI IfcAlignment [4]: The Alignment conceptual model in this LandInfra Standard was developed jointly with the bSI IfcAlignment conceptual model. The objective is that they be as similar as possible, with only minor differences due mostly to the respective environments in which these standards exist. This consistency should enable linking of geospatial and BIM data based upon linear location.

bSI IFCs for roads and railways: The intent is that the Road and Railway conceptual model in this LandInfra Standard be compatible with the forthcoming bSI IFC conceptual model. So far, only the proposed Korean road model [8] and Chinese railway models have been discussed in bSI and they may become regional PAS submittals instead of international bSI standards. Work on an international standard is commencing within bSI as IFC Roads and Railways. LandInfra does not intent to be as detailed as IFCs presumably will, so the specification of Road and Railway elements herein should not conflict with the eventual IFC model.

## 2. Conformance

This standard defines concepts for land and civil engineering infrastructure facilities.

Requirements for one standardization target type is considered:

- Land and Infrastructure encoding standards

Such encoding standards might include, for example, GML, IFC, SQL, XML, JSON, etc.

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site<sup>1</sup>.

In order to conform to this OGC™ conceptual standard, a standardization target shall choose to implement the core conformance class and any of the other conformance classes with their dependencies. Conformance classes are based on requirements classes which are specified in this standard.

All requirements classes and conformance classes described in this document are owned by the standard(s) identified.

---

<sup>1</sup> [www.opengeospatial.org/cite](http://www.opengeospatial.org/cite)

### 3. Normative References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

ISO 6707-1:2014, *Buildings and civil engineering works — Vocabulary — Part 1: General terms*

ISO 19103, *Geographic information – Conceptual schema language.*

ISO 19109:2015, *Geographic information – Rules for application schema*

ISO 19115, *Geographic information – Metadata*

ISO 19130, *Geographic information – Imagery sensor models for geopositioning*

OGC 01-101, *Abstract Specification Topic 1 - Feature Geometry*, (ISO 19107), May 10, 2001.

OGC 08-015r2, *Abstract Specification Topic 2 – Spatial Referencing by Coordinates*, (ISO 19111), April 27, 2010.

OGC 10-030, *Abstract Specification Topic 19: Geographic information - Linear referencing*, (ISO 19148), March 20, 2012.

OGC 10-004r3, *Abstract Specification Topic 20: Observations and Measurements*, November 10, 2010.

## 4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

For the purposes of this document, the following additional terms and definitions apply.

### 4.1 General Terms

#### 4.1.1

##### **assembly**

set of related **components** attached to each other

[ISO 6707-1:2014]

#### 4.1.2

##### **component**

distinct unit of **construction** or a **product** constructed or manufactured to serve a specific function or functions

Note 1 to entry: ISO 6707-1:2014 limits **components** to manufactured **products**

#### 4.1.3

##### **conceptual model**

model that defines concepts of a **universe of discourse**

[ISO 19103, 4.11, 4.23]

#### 4.1.4

##### **conformance class**

set of conformance test modules that must be applied to receive a single certificate of conformance

[OGC 08-131r3]

#### 4.1.5

##### **dataset**

identifiable collection of data

[ISO 19115]

4.1.6

**dimension**

extent in a given direction or along a given line, or a given angle

[ISO 1803:1997, 3.1]

4.1.7

**document**

information in permanent form which is approved by one or more signatures

Compares to ISO 19152:2012 4.1.21 source

4.1.8

**feature**

abstraction of real world phenomena

[ISO 19103, 4.16]

4.1.9

**gradient**

ratio of difference in **level** between two points to the horizontal distance between them

[ISO 6707-1:2014]

4.1.10

**height**

vertical **dimension** above a horizontal reference **level**

[ISO 6707-1:2014]

4.1.11

**height**

distance of a point from a chosen reference surface measured upward along a line perpendicular to that surface

Note 1 to entry: The surface is normally used to model the surface of the Earth.

[OGC 08-015r2]

4.1.12

**length**

one of two horizontal **dimensions**, normally the larger

Note 1 to entry: The other is **width**.

[ISO 6707-1:2014]

4.1.13

**level**

value of the vertical **dimension** of a point above or below a defined reference

[ISO 6707-1:2014]

4.1.14

**linear element**

1-dimensional object that serves as the axis along which **linear referencing** is performed

Note 1 to entry: Also known as curvilinear element.

EXAMPLES Feature, such as “road”; curve geometry; directed edge topological primitive, Alignment.

[OGC 10-030]

4.1.15

**party**

a person or organization that plays a role in a lawful process

Compares to ISO 19152:2012 4.1.13 party

4.1.16

**physical element**

any **component** defined within the spatial and functional context of a **facility**

4.1.17

**positioning element**

virtual element used to position, align, or organize **physical elements**

4.1.18

**product**

item manufactured or processed for incorporation in **construction works**

[ISO 6707-1:2014]

4.1.19

**professional**

person, generally with tertiary education, who through a license is bound to act according to the standards of a professional association

Note 1 to entry: In common usage, the term professional includes a much broader scope of individuals. Here, it only refers to those who are licensed, such as land surveyors.

4.1.20

**requirement class**  
**requirements class**

aggregate of all requirement modules that must all be satisfied to satisfy a conformance test class

[OGC 08-131r3]

4.1.21

**slope**

inclination of a plane surface to the horizontal

[ISO 6707-1:2014]

4.1.22

**survey mark**

physical object which by its form defines a point on the surface of the Earth

4.1.23

**universe of discourse**

view of the real or hypothetical world that includes everything of interest

[ISO 19101:2014, 4.1.38]

4.1.24

**user**

organization, person, animal, or object for which a **building** or other **construction works** is designed

[ISO 6707-1:2014]

4.1.25

**width**

one of two horizontal **dimensions**, normally the smaller

Note 1 to entry: The other is **length**.

[ISO 6707-1:2014]

## 4.2 Facility and Project Terms

### 4.2.1

#### **bridge**

**civil engineering works** that affords passage to pedestrians, animals, vehicles, and **services** above obstacles or between two points at a **height** above **ground**

[ISO 6707-1:2014]

### 4.2.2

#### **building**

**construction works** that has the provision of shelter for its occupants or contents as one of its main purposes, usually partially or totally enclosed and designed to stand permanently in one place

[ISO 6707-1:2014]

### 4.2.3

#### **civil engineering works**

**construction works** comprising a **structure**, such as a **dam**, **bridge**, **road**, **railway**, runway, utilities, **pipeline**, or **sewerage system**, or the result of operations such as dredging, **earthwork**, geotechnical processes, but excluding a **building** and its associated **siteworks**

Note 1 to entry: Associated **siteworks** are included in US civil engineering projects.

[ISO 6707-1:2014]

### 4.2.4

#### **construction**

assembled or complete part of **construction works** that results from work on-site

[ISO 6707-1:2014]

### 4.2.5

#### **construction work**

#### **construction**

activities of forming **construction works**

[ISO 6707-1:2014]

### 4.2.6

#### **construction works**

#### **construction**

everything that is constructed or results from construction operations

[ISO 6707-1:2014]

#### 4.2.7

##### **facility**

improvements of or on the **land** including buildings and civil engineering works and their associated siteworks

#### 4.2.8

##### **retaining wall**

**wall** that provides lateral support to the **ground** or that resists pressure from a mass of other material

[ISO 6707-1:2014]

#### 4.2.9

##### **site**

area of **land** or water where **construction work** or other development is undertaken

[ISO 6707-1:2014]

#### 4.2.10

##### **siteworks**

**construction works** or landscape work on **land** associated with, and adjacent to, **civil engineering works** or a **building**

[ISO 6707-1:2014]

#### 4.2.11

##### **structure**

**construction works** having an organized combination of connected parts designed to provide some measure of rigidity

[ISO 6707-1:2014]

#### 4.2.12

##### **track**

a small path mainly used by pedestrians

[OGC 08-007r1 CityGML]

Note 1 to entry: This is not the definition adopted by this standard. See instead 4.5.

#### 4.2.13

##### **tunnel**

horizontal or sloping underground enclosed way of some **length**

[ISO 6707-1:2014]

#### 4.2.14

##### **wall**

vertical **construction** that bounds or subdivides a space and usually fulfils a loadbearing or retaining function

[ISO 6707-1:2014]

### 4.3 Alignment Terms

#### 4.3.1

##### **alignment**

linear referencing system associated to linear **facilities** and their constructions, such as roads, railways, and bridges, used to position elements, such as road, railway or bridge elements or other **physical elements**, positioned along the alignment

#### 4.3.2

##### **horizontal alignment**

linear reference line projected onto the horizontal x/y plane in a Cartesian local engineering coordinate system

## 4.4 Road Terms

### 4.4.1

#### **carriageway**

#### **roadway**

part of the **road** or **highway** constructed for use by vehicular **traffic**, including auxiliary **traffic lanes**, passing places, and **lay-bys**

[ISO 6707-1:2014]

### 4.4.2

#### **central reserve**

#### **median**

area that separates the **carriageways** of a **road** with dual carriageways

[ISO 6707-1:2014]

### 4.4.3

#### **cycleway**

#### **bicycle path**

way or separated part of a **road** for use only by pedal cycles

[ISO 6707-1:2014]

### 4.4.4

#### **footpath**

way for the use of pedestrians

[ISO 6707-1:2014]

### 4.4.5

#### **footway**

#### **sidewalk**

#### **walkway**

portion of a **road** reserved exclusively for pedestrians

[ISO 6707-1:2014]

### 4.4.6

#### **gutter**

**channel** for collecting and draining rainwater

Note 1 to entry: ISO 6707-1:2014 adds “from a roof”

#### 4.4.7

**hard shoulder**  
**emergency lane**  
**service lane**

surfaced strip, adjacent to and abutting a **carriageway**, intended for use by vehicles in the event of difficulty or during obstruction of the carriageway

[ISO 6707-1:2014]

#### 4.4.8

**highway**  
**parkway**  
**freeway**

way over which the public has the right to pass, this right possibly being restricted to specific classes of **traffic**

[ISO 6707-1:2014]

#### 4.4.9

**kerb**  
**curb**

border, usually upstanding, at the edge of a **carriageway**, hard strip, **hard shoulder**, or **footway**

[ISO 6707-1:2014]

#### 4.4.10

**lay-by**  
**stopping lane**  
**emergency lane**

part of the **highway** set aside for vehicles to allow them to draw out of the **traffic lanes** and wait for short periods

[ISO 6707-1:2014]

#### 4.4.11

**motorway**  
**interstate highway**  
**freeway**  
**parkway**

limited access **road** with dual **carriageways** that is not crossed on the same **level** by other **traffic lanes**, for the exclusive use of certain classes of motor vehicles

[ISO 6707-1:2014]

#### 4.4.12

##### **pavement**

**road**, runway, or similar **construction** above the **subgrade**

[ISO 6707-1:2014]

#### 4.4.13

##### **road**

way mainly for vehicles

[ISO 6707-1:2014]

#### 4.4.14

##### **road marking**

line, symbol, or other mark on a **road** surface intended to regulate, warn, guide, or inform **users**

[ISO 6707-1:2014]

#### 4.4.15

##### **roundabout**

##### **rotary**

portion of a **road**, usually at a junction, on which **traffic** moves in one direction around a central element

[ISO 6707-1:2014]

#### 4.4.16

##### **traffic**

movement of vehicles, people, or animals along a way

[ISO 6707-1:2014]

#### 4.4.17

##### **traffic lane**

strip of **carriageway** intended to accommodate a single line of moving vehicles, frequently defined by **road markings**

[ISO 6707-1:2014]

#### 4.4.18

##### **vehicle restraint system**

##### **guardrail**

##### **barricade**

**structure** that provides a system of containment for errant vehicles so as to limit damage or injury

[ISO 6707-1:2014]

#### **4.4.19**

**verge**

**shoulder**

part of a **highway** alongside a **carriageway** and at approximately the same **level**, exclusive of **embankment** or **cutting** slopes

[ISO 6707-1:2014]

Note 1 to entry: It can include **footways** and **cycleways**.

## 4.5 Railway Terms

### 4.5.1

#### **cant**

The measurement of the difference in elevation between the outer **rail** and the inner **rail**.

Note 1 to entry: In curved track, it is usually designed to raise the outer rail, providing a banked turn, thus allowing trains to maneuver through the curve at higher speeds than would otherwise be possible if the surface was flat or level. It also helps a train steer around a curve, keeping the wheel flanges from pressing the rails, minimizing friction and wear.

### 4.5.2

#### **gauge**

spacing of the **rails** of a **track**, measured between the inner faces of the load-bearing **rails**

### 4.5.3

#### **rail**

a steel bar or continuous line of bars laid on the ground as one of a pair forming a railway track

[Oxford dictionary]

Note 1 to entry: Though this is the common usage definition, it is nonetheless included herein to clearly distinguish it from **railway** and **track**.

### 4.5.4

#### **railway**

#### **railroad**

national or regional transport system for guided passage of wheeled vehicles on rails

[ISO 6707-1:2014]

a track made of steel rails along which trains run

[Oxford dictionary]

### 4.5.5

#### **track**

**assembly** of rails, fastenings, and support, for passage of vehicles

[ISO 6707-1:2014]

## 4.6 Survey Terms

### 4.6.1

#### **antenna**

electronic equipment to transmit and/or receive electromagnetic waves cf. electronic distance meter (4.6.3), geodetic GPS receiver (4.6.5)

[ISO 9849-1:2000]

### 4.6.2

#### **digital field registration**

#### **digital recording**

#### **data logger**

device that immediately collects, records and recalls surveying data and other information

[ISO 9849-1:2000]

### 4.6.3

#### **electronic distance meter**

#### **EDM instrument**

electromagnetic distance meter instrument for directly measuring distances between the instrument and the sighting points, comprising a transmitter and a receiver

[ISO 9849-1:2000]

### 4.6.4

#### **electronic tacheometer**

#### **electronic stadia instrument**

#### **electronic tachymeter**

#### **total station**

#### **theodolite**

fitted with an electronic distance measuring device and with electronic scanning of the circles

[ISO 9849-1:2000]

### 4.6.5

#### **Global Positioning System**

#### **GPS**

U.S. Department of Defense navigation system of high accuracy

[ISO 9849-1:2000]

#### **4.6.6**

##### **Global Navigation Satellite System**

##### **GNSS**

using the GPS, GLONASS, Galileo or Beidou system

#### **4.6.7**

##### **imagery**

representation of phenomena as images produced by electronic and/or optical techniques

[ISO 19101-2:2008]

#### **4.6.8**

##### **inclination compensator**

##### **tilt compensator**

##### **dual-axis compensator**

##### **double-axis compensator**

device which automatically eliminates the influence of any levelling errors of a measuring instrument on the measured values

[ISO 9849-1:2000]

#### **4.6.9**

##### **level**

<geodetic instrument> instrument for measuring differences in height by establishing horizontal lines of sight, comprising as main components a telescope which can be rotated on a vertical axis and a facility for levelling the line of sight

[ISO 9849-1:2000]

#### **4.6.10**

##### **levelling staff**

##### **levelling rod**

##### **level rod**

straight bar of metal or wood with a scale on a flat face, primarily used to measure the vertical distance between a point and the horizontal line of sight of a level (4.6.9)

[ISO 9849-1:2000]

#### **4.6.11**

##### **measurement**

set of operations having the object of determining the value of a quantity

[ISO/TS 19101-2:2008, definition 4.20]

#### **4.6.12**

##### **observation**

act of measuring or otherwise determining the value of a property

[ISO/TS 19156, definition 4.10]

#### **4.6.13**

##### **observation procedure**

Method, algorithm or instrument, or system of these which may be used in making an observation

[ISO/DIS 19156, definition 4.11]

#### **4.6.14**

##### **observed value**

A value describing a natural phenomenon, which may use one of a variety of scales including nominal, ordinal, ratio and interval.

Note 1 to entry: The term is used regardless of whether the value is due to an instrumental observation, a subjective assignment or some other method of estimation or assignment.

[ISO 19156:2011]

#### **4.6.15**

##### **result**

##### **observation result**

An estimate of the value of some property generated by a known procedure

[ISO 19156:2011]

#### **4.6.16**

##### **sensor**

element of a measuring system that is directly affected by a phenomenon, body, or substance carrying a quantity to be measured

[ISO 19115-2]

#### **4.6.17**

##### **sensor data**

list of digital values produced by a sensor that represents estimated values of one or more observed properties of one or more features.

#### **4.6.18**

##### **theodolite**

##### **transit**

instrument for measuring horizontal directions or horizontal directions and vertical angles, whose main components are the horizontal circle and the vertical circle, the telescope and additional devices for reading the graduated circles and for setting up the vertical axis

[ISO 9849-1:2000]

**4.6.19**

**tacheometer**

**stadia theodolite**

**tachymeter**

**theodolite**

equipment designed for use in determining horizontal directions, distances and height differences

[ISO 9849-1:2000]

## 4.7 Land Feature Terms

### 4.7.1

#### **cut**

material excavated in bulk

[ISO 6707-1:2014]

### 4.7.2

#### **cut**

void that results from bulk **excavation** of material

[ISO 6707-1:2014]

### 4.7.3

#### **earthwork**

work of excavating (**cut**), or the raising (**fill**) or sloping of **ground**

[ISO 6707-1:2014]

### 4.7.4

#### **earthworks**

result of change of existing terrain

[ISO 6707-1:2014]

### 4.7.5

#### **embankment**

section of **earthworks**, often formed by **cut** or **fill**, where the finished **ground level** is above or below original **ground level** and whose **length** usually greatly exceeds its **width**

[ISO 6707-1:2014]

### 4.7.6

#### **excavation**

result of digging, lifting, and removing earth, **fill**, or other material from the **ground**

[ISO 6707-1:2014]

### 4.7.7

#### **fill**

material used for raising the **level** of the **ground**

[ISO 6707-1:2014]

#### 4.7.8

##### **finished ground level**

##### **finished grade**

level of paved area or surface of the **land** after improvements or **earthwork**

[ISO 6707-1:2014]

#### 4.7.9

##### **ground**

**soil**, rock, and **fill** existing in place prior to the execution of **construction works**

[ISO 6707-1:2014]

#### 4.7.10

##### **ground level**

##### **grade**

level at the surface of the **land**

[ISO 6707-1:2014]

#### 4.7.11

##### **land**

the surface of the Earth, the materials beneath, the air above and all things fixed to the soil

[ISO 19152:2012]

#### 4.7.12

##### **soil**

##### **earth**

mineral material that results from the weathering of rock or decay of vegetation

[ISO 6707-1:2014]

#### 4.7.13

##### **subgrade**

upper part of the **soil**, natural or constructed, that supports the loads transmitted by the overlying **structure** of a **road**, runway, or similar hard surface

[ISO 6707-1:2014]

## 4.8 Land Division Terms

### 4.8.1

#### **administrative division**

division of state territory according to political, judicial, or executive points of view.

Note 1 to entry: May include other man-made divisions, e.g. according to permitted land use, but excludes **drainage basins** and other divisions derived from material of the Earth.

### 4.8.2

#### **building part**

floor-related part of a multi-storage building, subdivided according to management and use by a lawful process.

Compares to ISO 19152:2012 4.1.6 building unit, and to IfcSpatialElement

### 4.8.3

#### **boundary**

set that represents the limit of an entity

Note 1 to entry: Boundary is most commonly used in the context of geometry, where the set is a collection of points or a collection of objects that represent those points. In other arenas, the term is used metaphorically to describe the transition between an entity and the rest of its domain of discourse.

[ISO 19107:2003, 4.4; ISO 19152:2012 4.1.3]

### 4.8.4

#### **condominium**

concurrent ownership of real property that has been divided into private and common portions

Note 1 to entry: Condominium unit owners must be members of a mandatory owners' association and engage in the maintenance of joint facilities according to a specified share.

### 4.8.5

#### **condominium scheme**

documentation of **interest in land** in terms of a **statement** concerning the **building parts** which make up a number of **condominium units** and the share of joint facilities of each **condominium unit**

#### 4.8.6

##### **condominium unit**

type of **property unit** which consist of one or more privately owned building parts and has a share in the commonly owned joint facilities, as specified in a **condominium scheme**

#### 4.8.7

##### **easement**

accessory interest in land which advantages dominant land or other beneficiary.

#### 4.8.8

##### **interest in land**

**ownership** or security towards real property

#### 4.8.9

##### **land**

area of earth's surface, excluding the oceans, usually marked off by natural or political boundaries, or boundaries of ownership

[ISO 6707-1:2014]

#### 4.8.10

##### **land**

the surface of the Earth, the materials beneath, the air above and all things fixed to the soil

[ISO 19152:2012]

#### 4.8.11

##### **land parcel**

contiguous part of the surface of the Earth (land and/or water) as specified through lawful process

#### 4.8.12

##### **liminal spatial unit**

**spatial unit** on the threshold between 2D and 3D representations

[ISO 19152:2012]

#### 4.8.13

##### **ownership (in land)**

includes the right to grant a lease, an **easement**, or a security interest and other lesser rights

Note 1 to entry: **Ownership in land** includes buildings and fixture on the land parcel, unless lawful process is performed in terms of establishment of **condominium units**.

#### **4.8.14**

##### **spatial unit**

contiguous geometrical entity, which is delimited and located on or close to the surface of the Earth through the bounding elements of its **boundary**.

Compares to ISO 19152:2012 4.1.23 spatial unit

#### **4.8.15**

##### **statement**

**document** used in lawful process

Examples include account, agreement, declaration, observation, and record

#### **4.8.16**

##### **strata title**

title to land that is not necessarily divided horizontally, such as in high-rise buildings or for mining rights.

[19]

#### **4.8.17**

##### **survey monument**

physical object which by its form defines a point on the surface of the Earth, which is stably located, and which is recorded in a **statement**.

Note 1 to entry: may carry a symbol of authority, which communicates its purpose

## 4.9 “Wet” Infrastructure Terms

### 4.9.1

#### **canal**

**channel** constructed to carry water, usually for navigation, but which can also be used for water power, irrigation, collecting rainwater run-off, or **drainage** of **surface water**

[ISO 6707-1:2014]

### 4.9.2

#### **channel**

open passage for conveying or containing water

[ISO 6707-1:2014]

### 4.9.3

#### **conduit**

**pipe, channel, or tunnel** used for conveying liquids or containing electric wires or cables

[ISO 6707-1:2014]

### 4.9.4

#### **culvert**

transverse drain or waterway **structure** under a **road, railway, or canal**, or through an **embankment**, in the form of a large **pipe** or enclosed **channel**

[ISO 6707-1:2014]

### 4.9.5

#### **dam**

barrier constructed to retain water in order to raise its level, form a reservoir, or reduce or prevent flooding

[ISO 6707-1:2014]

### 4.9.6

#### **drain**

**conduit**, usually underground, or **channel** which conveys **wastewater, surface water**, or other unwanted liquids

[ISO 6707-1:2014]

### 4.9.7

#### **drainage**

removal of surplus water

[ISO 6707-1:2014]

Note 1 to entry: In this standard, **drainage** shall refer only to **surface water**.

#### 4.9.8

##### **drainage basin watershed**

area from which all precipitation flows to a single point at a lower elevation

#### 4.9.9

##### **drainage system**

system of drains and ancillary works that conveys their contents to a cesspool, sewerage system, outfall, or other place of disposal

[ISO 6707-1:2014]

Note 1 to entry: In this standard, **drainage system** shall be used to denote **drainage** of **surface water** only.

#### 4.9.10

##### **land drainage**

system of **conduits**, **structures**, and **embankments** required to control water **levels** and to protect urban and agricultural **land** from flooding by either fresh or salt water, or to alleviate such flooding

[ISO 6707-1:2014]

#### 4.9.11

##### **manhole**

opening fitted with a removable cover, which permits entry of a person to a **pipeline** or closed vessel

[ISO 6707-1:2014]

#### 4.9.12

##### **pipe**

circular **tube** through which fluid can flow

[ISO 6707-1:2014]

#### 4.9.13

##### **pipeline**

long continuous line of **pipes**, including ancillary equipment, used for transporting liquids or gases

[ISO 6707-1:2014]

#### 4.9.14

##### **service**

system for conveying water, gas, warm air, or electricity, or that provides water, gas, oil, or air to or within a **construction works** or removes waste from it

[ISO 6707-1:2014]

#### 4.9.15

##### **sewer**

**pipeline** or other **construction**, usually underground, which conveys unwanted liquids

[ISO 6707-1:2014]

#### 4.9.16

##### **sewerage system**

##### **sewage system**

system of **sewers** and ancillary works that conveys the contents to a sewage treatment works or other place of disposal

[ISO 6707-1:2014]

#### 4.9.17

##### **surface water**

water that flows over, rests on, or drains from the surface of **buildings**, other **structures**, or the **ground**

[ISO 6707-1:2014]

#### 4.9.18

##### **tube**

##### **pipe**

hollow **product**, usually formed by a continuous process to a definite cross-section, which is small in relation to its **length**

[ISO 6707-1:2014]

#### 4.9.19

##### **wastewater**

##### **sewage**

water discharged after being used in a household or in a process, or produced by a process, other waters in a combined system and water that has infiltrated a **sewerage system**

[ISO 6707-1:2014]

#### **4.9.20**

##### **water distribution system**

water supply system, including **pipes**, valves, and pumps

#### **4.9.21**

##### **wet infrastructure**

infrastructure improvements which include **water distribution system**, storm **drainage systems**, and **wastewater** systems

## 5. Conventions

### 5.1 Abbreviations

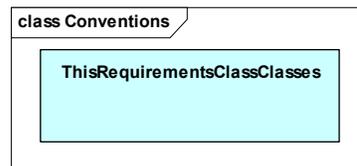
In this document the following abbreviations and acronyms are used or introduced:

bSI	buildingSMART International
GML	Geography Markup Language
IFC	Industry Foundation Class
ISO	International Organization for Standardization
LADM	Land Administration Domain Model
OGC	Open Geospatial Consortium
UML	Unified Modeling Language

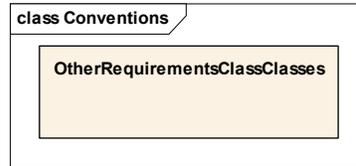
### 5.2 UML Package and Class Diagrams

The Conceptual Model contained within this document uses UML2 in accordance with well-established conventions adopted by the OGC and ISO TC211. Boxes shown within the UML Package and Class Diagrams are color coded as follows:

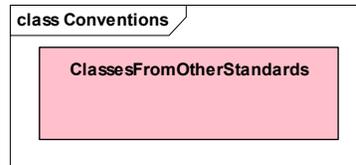
Blue represents Classes defined in this standard (internal) that are part of the Requirements Class being specified:



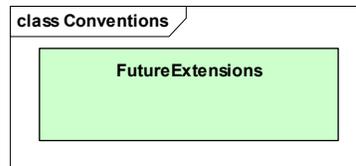
Beige represents Packages and Classes defined in this standard (internal) that are part of another Requirements Class other than the one being specified:



Pink represents Packages and Classes defined in other (OGC or ISO) standards (external):



Green represents possible Packages and Classes in future extensions to this standard:



Other colors may be specified as appropriate.

### 5.3 Requirements

When referred to in a Requirement or Requirements Class, the boxes contained in UML figures may all be called “Classes” even if they are data types, enumerations, code lists, unions etc. Because this is a Conceptual Model, they all should be interpreted to be concepts (see clause 6).

When a Requirement states that “The Requirements Class Classes shown in blue in Figure [nn] shall be provided for by the encoding in a manner consistent with the encoding.”, unless specified otherwise, this means that the encoding must support:

- a) all classes shown as blue boxes (this Requirements Class) in the figure
- b) all attributes, attribute cardinalities, and attribute data types of these classes (usually shown in subsequent figures)

- c) all associations, navigation, roles, and role cardinalities connecting to the blue classes
- d) all classes shown as beige boxes (another Requirements Class) in the figure connected to the blue box classes by association or used as attribute data types
- e) all classes shown as pink boxes (another Standard) in the figure connected to the blue box classes by association or used as attribute data types

The URI base for this standard is <http://www.opengis.net/spec/landinfra/1.0>. All URIs of Requirements Classes, Requirements, and Conformance Classes are relative to this base.

## 6. Conceptual Modeling (informative)

ISO 19101 [9] defines universe of discourse to be a view of the real or hypothetical world that includes everything of interest. That standard then defines conceptual model to be a model that defines concepts of a universe of discourse.

The scope of this LandInfra Standard establishes the limits of the universe of discourse for this Standard. The next task is to discover and standardize the concepts within this scope. LandInfra will potentially support numerous diverse application software packages covering multiple disciplines and facility life cycle phases. Each conceivably can have its own universe of discourse and a corresponding set of concepts.

The goal of this LandInfra Standard is to establish and document a common set of concepts that spans the applications supported. This does not attempt to redefine application concepts, but merely present a common set of concepts from and to which their concepts can be understood and mapped.

GML and IFC encodings are planned and other encodings are anticipated. Each encoding addresses a specific information community and set of application software packages. However, with the increasing desire to share information between communities and applications having a common conceptual model across all of these encodings is highly advantageous.

An added benefit of the development of a conceptual model results from the rigor involved in achieving consensus. After numerous iterations, the end result is consistent, cohesive, and complete. Updating a conceptual model is far easier than rewriting software code. Further, the iterations help to flesh out details as well as to unearth differences in individual conceptualizations.

Perhaps the greatest benefit of the standards activity is the ability to communicate the resultant model. This is in part due to using a standardized conceptual modelling language like UML and the agreed OGC and ISO TC211 conventions for using UML. The eventual outcome of being able to provide formal documentation for what is meant by each concept is invaluable in understanding the subsequent encodings and applications.

This will be the first OGC conceptual model standard, without accompanying encodings. Yet the model is presented in a manner consistent with the formalisms adopted for writing OGC standards. This standard follows the OGC Specification Model standard for modular specifications [10] and is consistent with the OGC Naming Authority conventions and recommendations. The targets of this Standard are the encoding standards which will follow and not the application software that will implement these encodings. Requirements for the encodings are explicit and grouped into Requirements

Classes. Accompanying Conformance Classes are included to determine if an encoding conforms to the conceptual model.

UML has been used as the conceptual modeling language in this LandInfra Standard. Class Diagrams have been created and inserted as Figures. The boxes in these diagrams (officially “Classifiers” in UML) typically represent classes, data types, enumerations, code lists, unions, etc. and this terminology is used throughout the Standard. However, since this is a Conceptual Model, these should all be interpreted to be “concepts”. For each Requirements Class, an introductory diagram is included which contains all of the concepts relevant to that Requirements Class. However, the boxes are simplified by suppressing attributes. These attributes are provided in a series of context diagrams which follow, each focusing on a particular set of concepts in the Requirements Class.

Though redundant with the UML diagrams, all of the class attributes are repeated in the document text, including attribute definitions not visible in the diagrams. If these differ, the UML takes precedence. Because association roles behave similar to attributes, they appear at the end of the textual attribute listing as if they were attributes. The cardinality of the association is depicted as the attribute cardinality and the associated class as the data type.

## **7. UML Conceptual Model (normative)**

### **7.1 Structural Overview of Requirements Classes**

The Requirements Classes for this standard are structured as UML Packages in Figure 1. Below is a brief summary of the function of each of these Requirements Classes.

#### **LandInfra**

LandInfra is the core Requirements Class and is the only mandatory Requirements Class. This class contains information about the Land and Infrastructure dataset that can contain information about facilities, land features, land division, documents, survey marks, surveys, sets, and feature associations. LandInfra also contains the definition of types common across other Requirements Classes, such as the Status CodeList.

#### **Facility**

Facilities include collections of buildings and civil engineering works and their associated siteworks. The Facilities Requirements Class includes the breakdown of facilities into discipline specific facility parts and introduces the notion of elements which make up these parts. The Facilities Requirements Class only provides general support for facilities themselves, allowing subsequent Requirements Classes to focus on specific types of the parts that make up facilities, such as road and railway. This Requirements Class is optional in order to allow for the condition where all of the LandInfra dataset information is not facility related, such as one containing only survey or land division information.

#### **Project**

A project is an activity related to the improvement of a facility, including design and/or construction. This class may be for the creation, modification, or elimination of the entire facility or a part of the facility. The Project Requirements Class includes information about projects and their decomposition into project parts. In order to allow for the condition where none of the LandInfra dataset information is project related, this Requirements Class is optional.

#### **Alignment**

An alignment is a positioning element which provides a Linear Referencing System for locating physical elements. The Alignment Requirements Class specifies how an alignment is defined and used.

#### **Road**

The Road Requirements Class supports those use cases in which a designer wishes to exchange the output of the design with someone who is likely to use it for purposes other than completing the road design. Consequently, the Road Requirements Class includes several alternative ways for representing a design such as with 3D RoadElements, 3D StringLines (aka profile views, longitudinal breaklines, long sections), and 3D surfaces and layers, as well as collections of these.

### **RoadCrossSection**

The RoadCrossSection Requirements Class extends the Road Requirements Class by adding the 2D CrossSection alternative way of representing a design, as well as collections of these.

### **RoadDesign (future work)**

The RoadDesign Requirements Class will support designer to designer information interchange, such as would exist when a designer other than the original designer takes over the design process to complete the design. It is anticipated that the (future proposed) RoadDesign Requirements Class will cover design information in support of those use cases which involve the exchange of design information between designers. It therefore would include DesignTemplates and SuperelevationEvents.

### **Railway**

The Railway Requirements Class supports those use cases where a designer wishes to exchange the output of the railway design with someone who is likely to use it for purposes other than further design. Consequently, the Railway Requirements Class covers design output such as 3D railway elements and track geometry including superelevation (cant).

### **Survey**

The Survey Requirements Class is the main survey class and provides a framework for information about observations, processes and their results collected during survey work. The content of this package is similar to the OGC Sensor Model Language (SensorML). The main reason not to use the SensorML standard for this topic is to allow the observation and processes structured in a compact way similar to the LandXML format. Due to the high number of classes the Survey Package was split into different parts, Equipment, Observations and SurveyResults.

### **Equipment**

In the Equipment Requirements class all information about the processes and the sensors is available that had been used for the determination of an observation.

### **Observations**

Observations Requirements class is the package containing all information about the raw observations and the measurements observed during survey work. The raw observations are needed to enable later reprocessing or reporting of resulting properties of the observed feature of interest.

### **Survey Results**

The SurveyResults Requirements Class contains the observed property of a feature of interest. Using sampling features from the Observation & Measurements (O&M) standard, the dependencies between the observation acts and the results are realized.

### **LandFeature**

Features of the land, such as naturally occurring water features and vegetation are specified in the LandFeature Requirements Class as land features. Also included are models of the land surface and subsurface layers. Improvements to the land such as the construction of an embankment or the planting of landscape material are considered to be part of Site Facilities in the Facility Requirements Class.

### **LandDivision**

Land can be divided up into land divisions. These can either be public (political, judicial, or executive) or private in nature. The former are administrative divisions and the latter are interests in land. Both of these are specified in the LandDivision Requirements Class, though condominium interests in land are specified in a separate, Condominium Requirements Class.

### **Condominium**

A condominium denotes concurrent ownership of real property that has been divided into private and common portions. The Condominium Requirements Class includes information about condominium units, buildings and schemes.

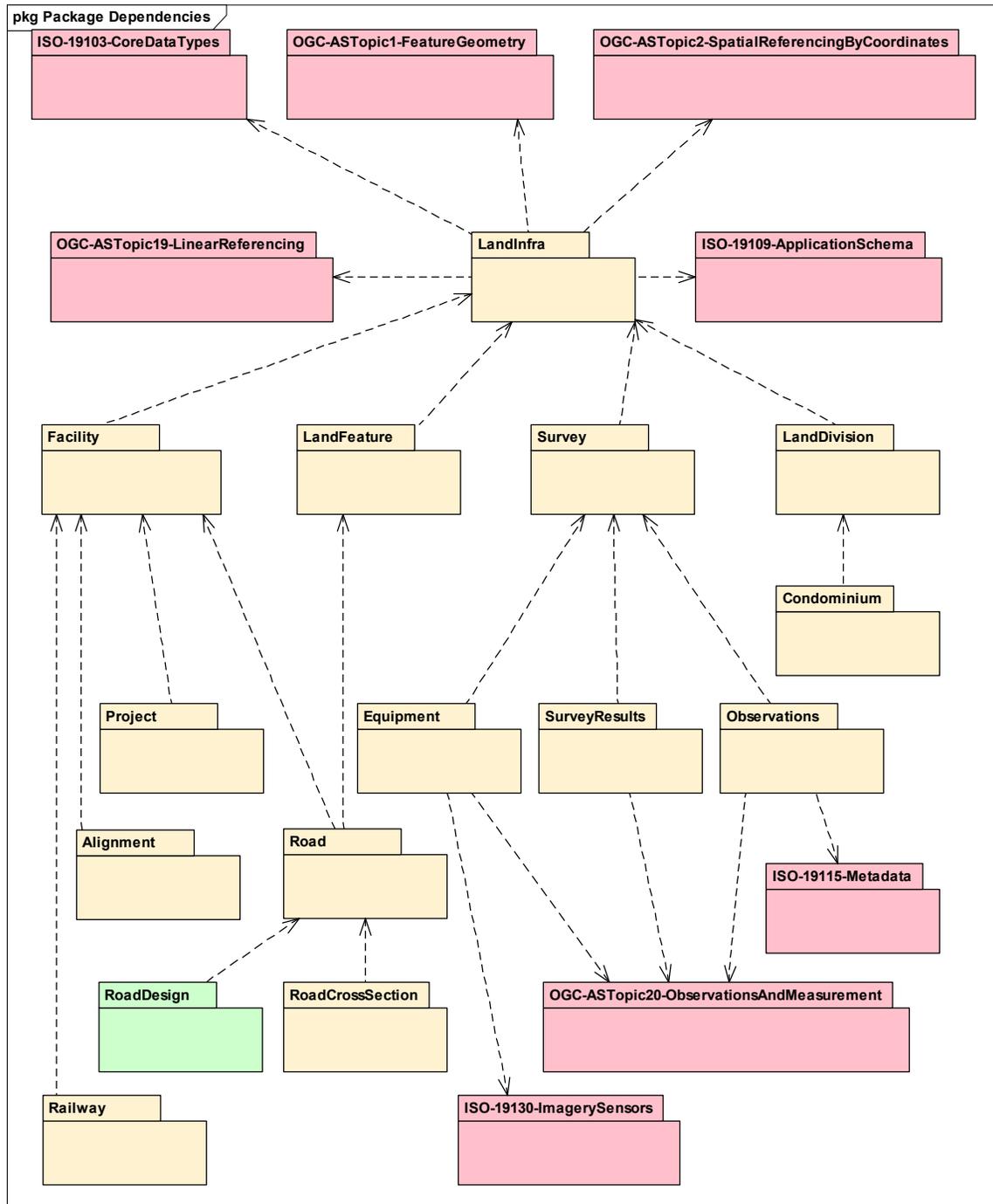


Figure 1. Requirements Classes as UML Packages with their dependencies

Figure 1 also shows (external) OGC and ISO standards as Packages on which Requirements Classes in this Standard depend. Below is a brief summary of the function of each of these Standards.

### **OGC-ASTopic1-FeatureGeometry**

Provides most of the geometry types (e.g., Point, LineString, Polygon) used for spatial representations in this Standard.

### **OGC-ASTopic2 - SpatialReferencingByCoordinates**

Defines Coordinate Reference Systems.

### **OGC-ASTopic19-LinearReferencing**

Defines the linear referencing concepts (e.g., linear element, distance along, Linear Referencing Methods) used for linearly referenced locations in this Standard.

### **OGC-ASTopic20-ObservationsAndMeasurement**

Defines a conceptual schema for observations, and for features involved in sampling when making observations.

### **ISO-19103-CoreDataTypes**

Provides the core data types (e.g., CharacterString, Date, Boolean) used in this Standard.

### **ISO-19109-ApplicationSchema**

Defines the General Feature Model upon which this Standard is based.

### **ISO-19115-Metadata**

Identifies the metadata to describe digital geographic data.

### **ISO-19130-Imagery sensor models for geopositioning**

Identifies the information required to determine the relationship between the position of a remotely sensed pixel in image coordinates and its geoposition.

## 7.2 Core Requirements Class: LandInfra

Core Requirements Class	
<b>/req/land-infra</b>	
Target type	Encoding of the conceptual model
Name	LandInfra
Dependency	<a href="urn:iso:is:iso:19103:clause:7">urn:iso:is:iso:19103:clause:7</a>
Dependency	<a href="http://www.opengis.net/doc/AS/Topic1">http://www.opengis.net/doc/AS/Topic1</a>
Dependency	<a href="http://www.opengis.net/doc/AS/Topic19">http://www.opengis.net/doc/AS/Topic19</a>
Dependency	<a href="urn:iso:is:iso:19109:req:uml:feature">urn:iso:is:iso:19109:req:uml:feature</a>
Dependency	<a href="http://www.opengis.net/doc/AS/Topic2">http://www.opengis.net/doc/AS/Topic2</a>
Requirement	<a href="#">/req/land-infra/dataset</a>
Requirement	<a href="#">/req/land-infra/subjects</a>
Requirement	<a href="#">/req/land-infra/classes</a>
Requirement	<a href="#">/req/land-infra/crs</a>
Requirement	<a href="#">/req/land-infra/featureID</a>
Requirement	<a href="#">/req/land-infra/19103</a>
Requirement	<a href="#">/req/land-infra/topic-1</a>
Requirement	<a href="#">/req/land-infra/topic-19-core</a>
Requirement	<a href="#">/req/land-infra/topic-19-extensions</a>
Requirement	<a href="#">/req/land-infra/19109</a>

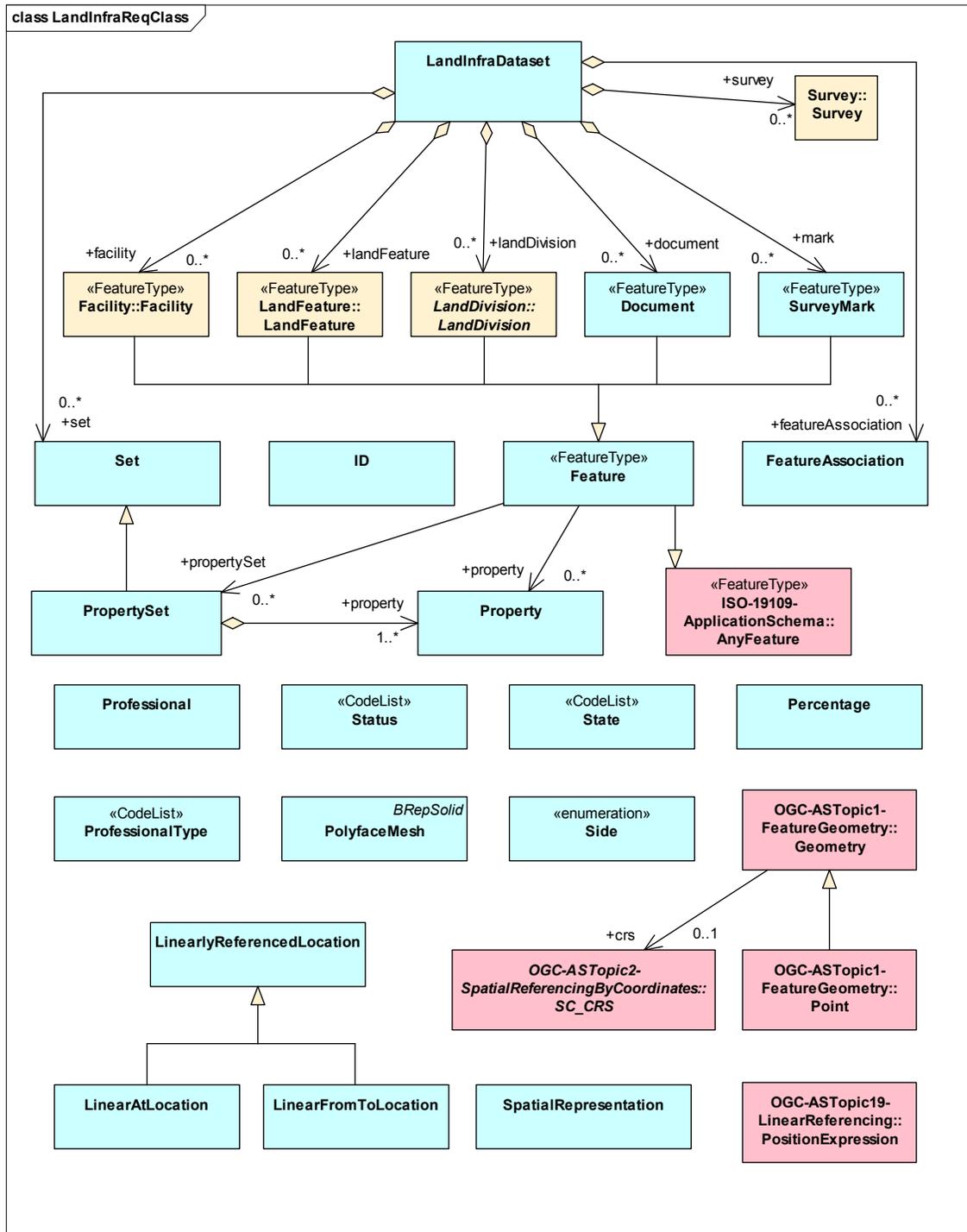


Figure 2. LandInfra Core Requirements Class

LandInfra is the core requirements class, as shown in Figure 2. The root class, LandInfra, specifies the Land and Infrastructure dataset.

<b>Requirement</b>	<b><a href="#">/req/land-infra/dataset</a></b>  A Land and Infrastructure encoding shall specify a LandInfra dataset in whatever format that is appropriate to that encoding (e.g., an XML document for a GML encoding).
--------------------	--

The dataset may contain any number of occurrences of each of the following: facility, land feature, land division, survey, document, survey mark, set, and feature association. The first four are subject areas addressed by explicit requirements classes.

<b>Requirement</b>	<b><a href="#">/req/land-infra/subjects</a></b>  A Land and Infrastructure encoding shall specify which of the LandInfra subjects it supports: Facility, LandFeature, LandDivision, and Survey.
--------------------	---

The LandInfra Requirements Class contains classes (Figure 3) and data types (Figure 4) common across multiple subjects.

## 7.2.1 LandInfra Requirements Class Classes

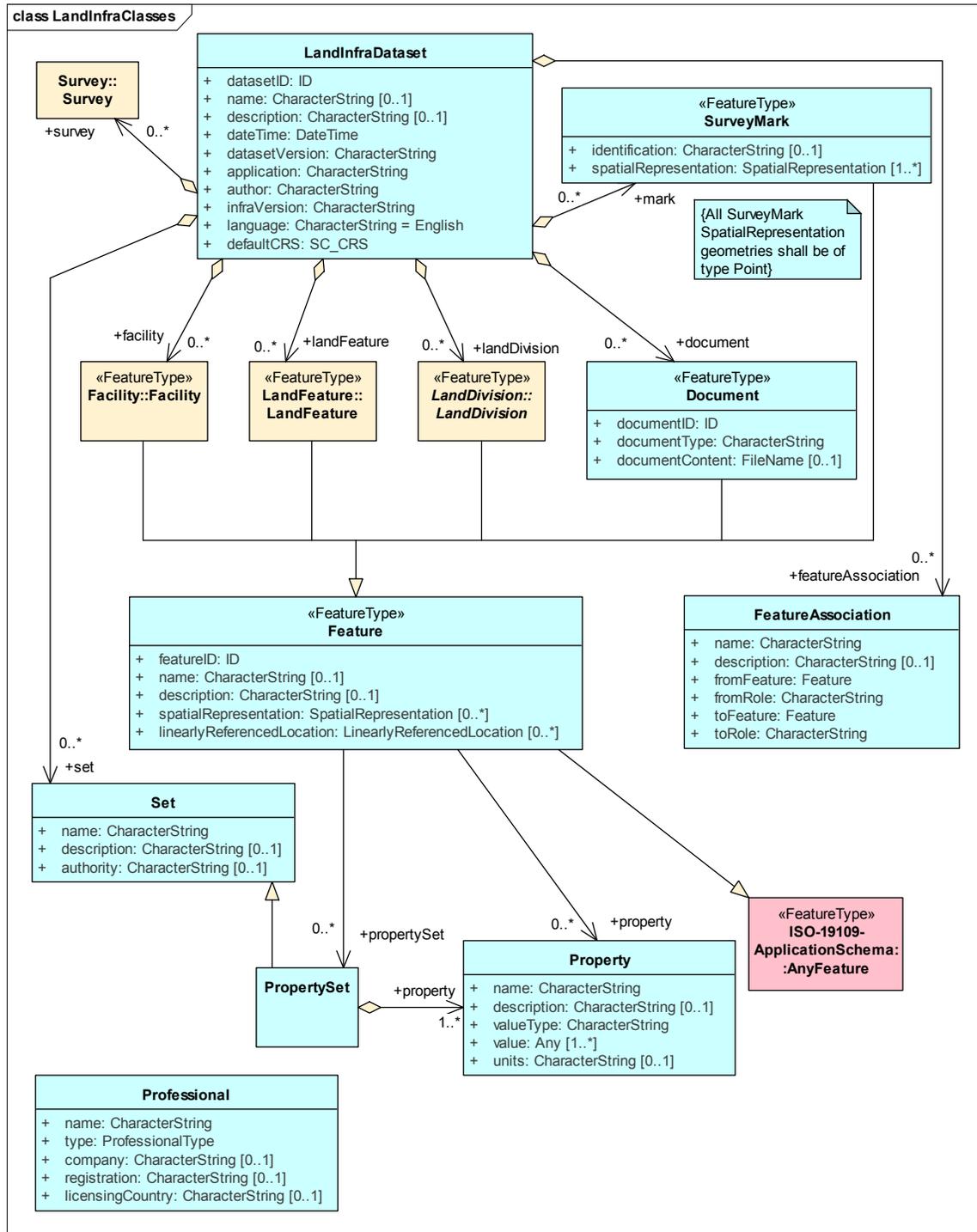


Figure 3. LandInfra Requirements Class Classes

<b>Requirement</b>	<p><b><a href="#">/req/land-infra/classes</a></b></p> <p>The LandInfra Requirements Class Classes shown in blue in Figure 2 shall be provided by the encoding in a manner consistent with the encoding.</p>
--------------------	---

### 7.2.1.1 LandInfra Dataset Class

The LandInfraDataset Class contains header information (metadata) about the LandInfra dataset:

- datasetID:** user defined unique ID used to identify a LandInfra dataset
- name:** optional user defined name of the dataset
- description:** optional description of the dataset
- dateTime:** date and time that the dataset was created and therefore the point-in-time for which the data is valid
- datasetVersion:** version number of the dataset
- application:** software application (including version) used to generate the dataset
- author:** person or organization which created this dataset
- infraVersion:** version number of the Land and Infrastructure encoding
- language:** language used in the dataset for CharacterString data types, default = English
- defaultCRS:** default coordinate reference system which is used for all spatial representations within the dataset except where overridden by the Alignment2DHorizontal.crs or any individual SpatialRepresentation.geometry value
- facility:** any number of Facilities
- landFeature:** any number of LandFeatures
- landDivision:** any number of LandDivisions
- document:** any number of Documents
- mark:** any number of SurveyMarks

**survey:** any number of Surveys

**set:** any number of Sets

**featureAssociation:** any number of associations between Features

<b>Requirement</b>	<b>/req/land-infra/crs</b> An encoding shall support coordinate reference systems in accordance with OGC Abstract Specification Topic 2, Spatial Referencing by Coordinates.
--------------------	---

### 7.2.1.2 Feature Class

The LandInfra Feature class is a specialization of the ISO 19109 AnyFeature Class. (See 7.2.7, ISO 19109 Application Schema below.) Additional Feature Types appropriate to the supported LandInfra subject areas will be further specializations of the LandInfra Feature Class. They will therefore inherit the following LandInfra Feature Class attributes:

**featureID:** user defined ID used to identify the feature instance, unique within the LandInfra dataset or globally unique with the inclusion of an ID.scope value

**name:** optional user defined name of the feature

**description:** optional description of the feature

**spatialRepresentation:** the optional spatial representation (geometry and location) of the feature specified here with a value type of SpatialRepresentation. Specific subtypes of Feature may make spatialRepresentation mandatory and may specify a more restrictive set of allowable subtype(s) of geometry (see 7.2.5).

**linearlyReferencedLocation:** the optional linearly referenced locations of the feature expressed as specified as a LinearlyReferencedLocation in OGC Abstract Specification Topic 19, Linear Referencing (see 7.2.6)

**property:** any number of Properties

**propertySet:** any number of PropertySets

FeatureID shall be unique across all Feature instances in the LandInfra dataset. Feature subtypes typically also have an ID, unique across that subtype. In either case, the ID can be made to be globally unique with the inclusion of an ID.scope value.

<b>Requirement</b>	<p><b>/req/land-infra/featureID</b></p> <p>An encoding shall specify if it supports ID at the feature level, ID at the Feature sub-type level, or both.</p>
--------------------	---

### 7.2.1.3 Document

A LandInfra dataset can contain information about any number of Documents. It has the following attributes:

**documentID:** user defined unique ID used to identify the document, unique within the dataset or globally unique with the inclusion of an ID.scope value

**documentType:** the type of document, e.g., Condominium scheme

**documentContent:** optional filename of a file containing the actual document

### 7.2.1.4 Survey Mark

A SurveyMark is a physical object which by its form defines a point on the surface of the Earth and which is stable during surveying operations.

**identification:** optional, user-defined identification

**spatialRepresentation:** one or more SpatialRepresentations having geometry of type Point

### 7.2.1.5 Property

An individual end user defined property (attribute) added to a Class.

**name:** name of the Property

**description:** optional description of the Property

**valueType:** the data type of the value of this Property

**value:** the actual Property value(s)

**units:** optional unit of measure of this Property

Not to be confused with “Property” in terms of real estate or immobile property which is addressed in 7.10.1, LandDivision.

#### 7.2.1.6 Set

Set is a general purpose collection class used to group instances of a particular class and to provide information about that grouping. Subtypes of Set are defined based on the types of things they can contain. Set contains the following attributes:

**name:** name of the Set, unique within the Set subtype

**description:** optional description of the set and/or its contents

**authority:** optional authority for which the set is defined

#### 7.2.1.7 PropertySet

PropertySet is a subtype of the Set general purpose collection class used to group instances of a particular class (in this case Property) and to provide information about that grouping. PropertySet has the following attribute:

**property:** one or more Properties in the set

By having both Feature Property and PropertySet allows for the specification of both individual properties and/or groups of properties for the Feature.

#### 7.2.1.8 Professional

In this LandInfra Standard, Professional is a person who has acquired advanced knowledge and is a member of an association with the obligation to adhere to ethical standards and to act with probity in relation to its clients and to the public. The association and the state establishes entry criteria, issues licenses, and often arranges for disciplinary tribunals [11].

Professional has the following attributes:

**name:** the name of the professional person

**type:** the name of the professional discipline

**company:** the optional name of the company through which the professional offers service

**registration:** an optional key or identification regarding the license(s) or similar authorization by the competent authority. Reference to the professional's case identification is specified in section 7.10.4, Statement, caseID.

**licensingCountry:** optional 3 character ISO country code for the country of the authority who issued the license

### 7.2.1.9 FeatureAssociation

The general purpose FeatureAssociation allows an implementation to dynamically create binary associations between any two Features in support of the Generalized Feature Model in ISO 19107. If a specific association has already been identified as a requirement in LandInfra, it is accommodated by a Feature to Feature association within a specific Requirements Classes. The general FeatureAssociation class has the following attributes:

**name:** name of the FeatureAssociation

**description:** optional description of the FeatureAssociation

**fromFeature:** the Feature considered to be the “from” Feature

**fromRole:** the role that the fromFeature plays in the FeatureAssociation

**toFeature:** the Feature considered to be the “to” Feature

**toRole:** the role that the toFeature plays in the FeatureAssociation

## 7.2.2 LandInfra Requirements Class Types

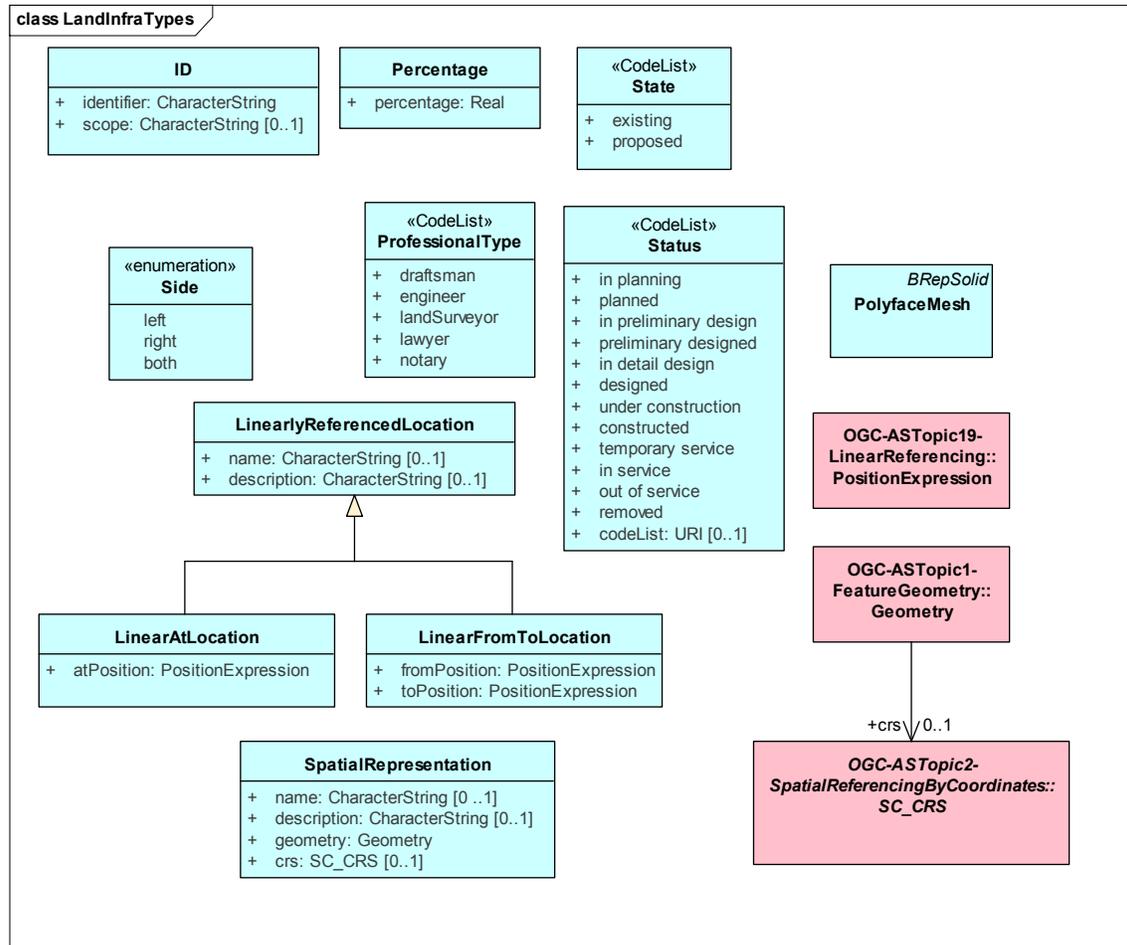


Figure 4. LandInfra Requirements Class Types

### 7.2.2.1 ID Data Type

The ID data type is used to uniquely identify an instance of a Class:

**identifier**: user defined ID, unique within the LandInfra dataset or globally unique with the inclusion of an ID.scope value.

**scope**: optional identifier used to make the ID globally unique. If not specified, the scope is assumed to be the dataset.

### **7.2.2.2 State Code List**

State defines whether an object is existing or proposed. The State code list values are therefore:

**existing**

**proposed**

### **7.2.2.3 Status Code List**

Status defines where the object of interest is within the facility life cycle. Though history can be documented across multiple successive LandInfra datasets, each LandInfra dataset is a single point-in-time snapshot. Therefore, each object can have at most one status value in a single LandInfra dataset. The code list values for Status are:

**in planning**

**planned**

**in preliminary design**

**preliminary designed**

**in detail design**

**designed**

**under construction**

**constructed**

**temporary service**

**in service**

**out of service**

**removed**

### **7.2.2.4 Side Enumeration**

The Side enumerated values are:

**left**

**right**

**both**

#### 7.2.2.5 Percentage Data Type

The Percentage data type is used to express a percentage (per hundred) as a real number. Therefore, a cross slope with percentage = 8.0 would be equivalent to a slope or gradient of .08 (8/100).

**percentage:** the real number value which is to be divided by 100 to obtain a decimal fraction of the whole

#### 7.2.2.6 Professional Data Type

Land development and infrastructure projects are assumed to engage engineers, land surveyors, lawyers, and notaries, subject to jurisdiction-specific practices. When in charge of preparing a document, they are typified as draftsman. The professions are characterized through reference to international confederations, where possible.

**draftsman:** the Professional who prepared (drafted) a Statement

**engineer:** American Society of Civil Engineers, Engineers Australia, European Federation of National Engineering Associations, and others

**landSurveyor:** The International Federation of Surveyors (FIG) is a confederation of national member associations

**lawyer:** The International Bar Association (IBA) provides support for national bar associations and law societies

**notary:** National chambers of notaries are members of the International Union of Notaries (UINL)

#### 7.2.2.7 SpatialRepresentation

SpatialRepresentation specifies a geometry and location using spatial data types in accordance with OGC Abstract Specification Topic 1, Geometry Types with Extensions (see 7.2.5). A Feature may have multiple spatial representations in accordance with ISO 19109. When this occurs, a name and description attribute are provided to differentiate between the multiple SpatialRepresentations.

**name:** optional name used to distinguish a SpatialRepresentation, unique within the set of spatialRepresentation values of a particular Feature

**description:** optional description of the SpatialRepresentation

**geometry:** the geometry/location of a Feature for this SpatialRepresentation

**crs:** optional Coordinate Reference System for the geometry appropriate only if the crs is different than the LandInfraDataset.defaultCRS value

#### 7.2.2.8 LinearlyReferencedLocation

LinearlyReferencedLocation specifies a location whose position is specified using linear referencing in accordance with OGC Abstract Specification Topic 19, Linear Referencing (see 7.2.6). It is specified with either a single “at” location or a pair of “from” and “to” locations.

**name:** optional name used to distinguish a LinearlyReferencedLocation, unique within the set of linearlyReferencedLocation values of a particular Feature

**description:** optional description of the LinearlyReferencedLocation

#### 7.2.2.9 LinearAtLocation

Single LinearlyReferencedLocation specifying where something is located along a linear element as a single “at” location.

**atPosition:** the linearly referenced “at” location

#### 7.2.2.10 LinearFromToLocation

Pair of LinearlyReferencedLocations specifying where something is located along a linear element, specifying its start and end location.

**fromPosition:** the linearly referenced “from” location where the located item begins

**toPosition:** the linearly referenced “to” location where the located item ends

### 7.2.3 LandInfra Requirements Class Associated Classes

The LandInfra Class aggregates any number of instances of the subject classes: facility, land feature, land division, and survey. Each of these are specified in their own Requirements Class(es).

## 7.2.4 ISO 19103 Core Data Types

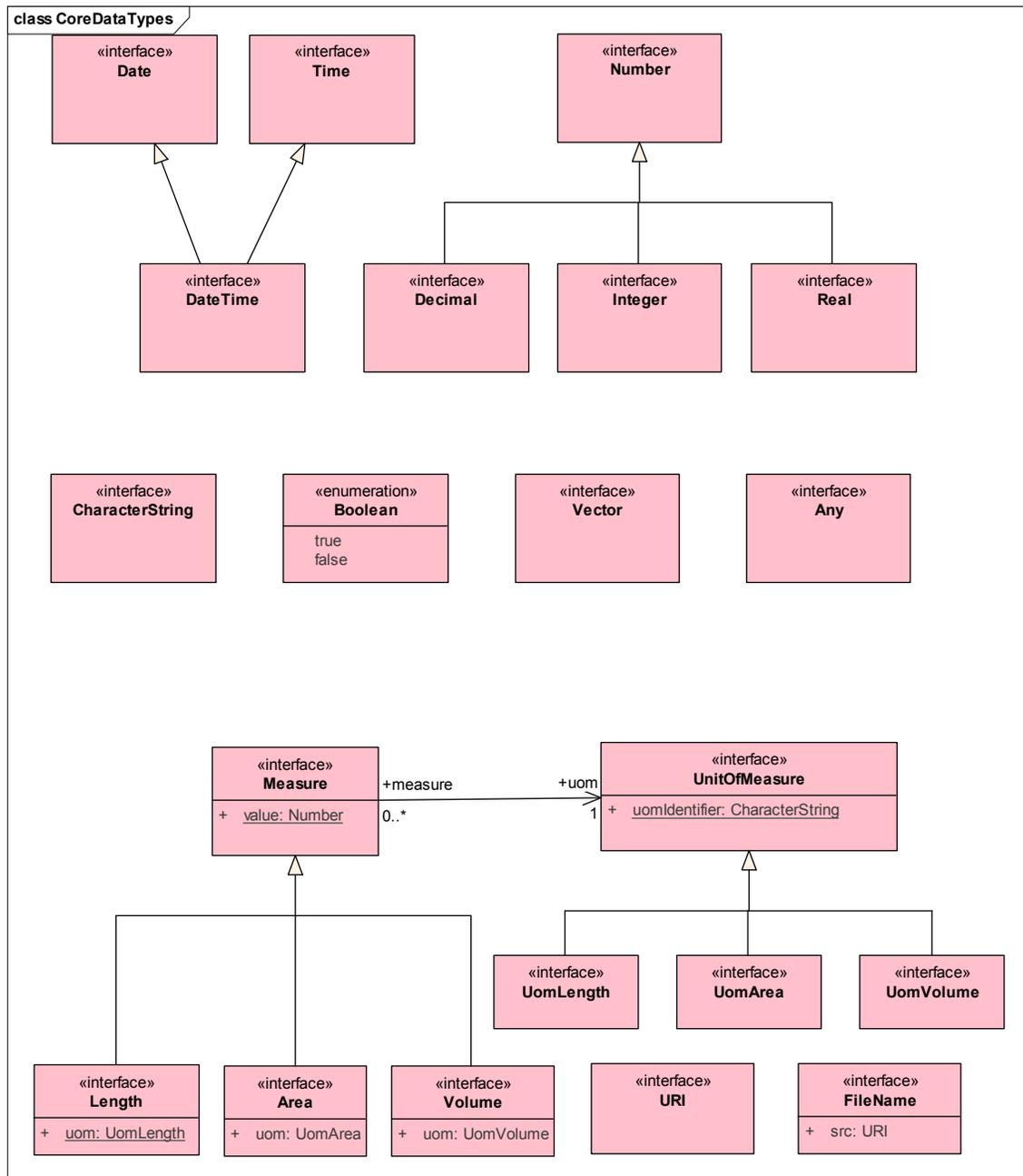


Figure 5. ISO 19103 Core Data Types

<b>Requirement</b>	<b><a href="#">/req/land-infra/19103</a></b> A Land and Infrastructure encoding shall support the core data types specified in ISO 19103 Clause 7 and shown in pink in Figure 5 that are appropriate to the supported subject area(s).
--------------------	---

The DateTime data type from ISO 19103 presupposes ISO 8601 Gregorian Calendar/UTC. If other temporal reference systems are to be supported, then TM\_Position from ISO 19108 would be more appropriate to use.

## 7.2.5 OGC Abstract Specification Topic 1 – Geometry Types with Extensions

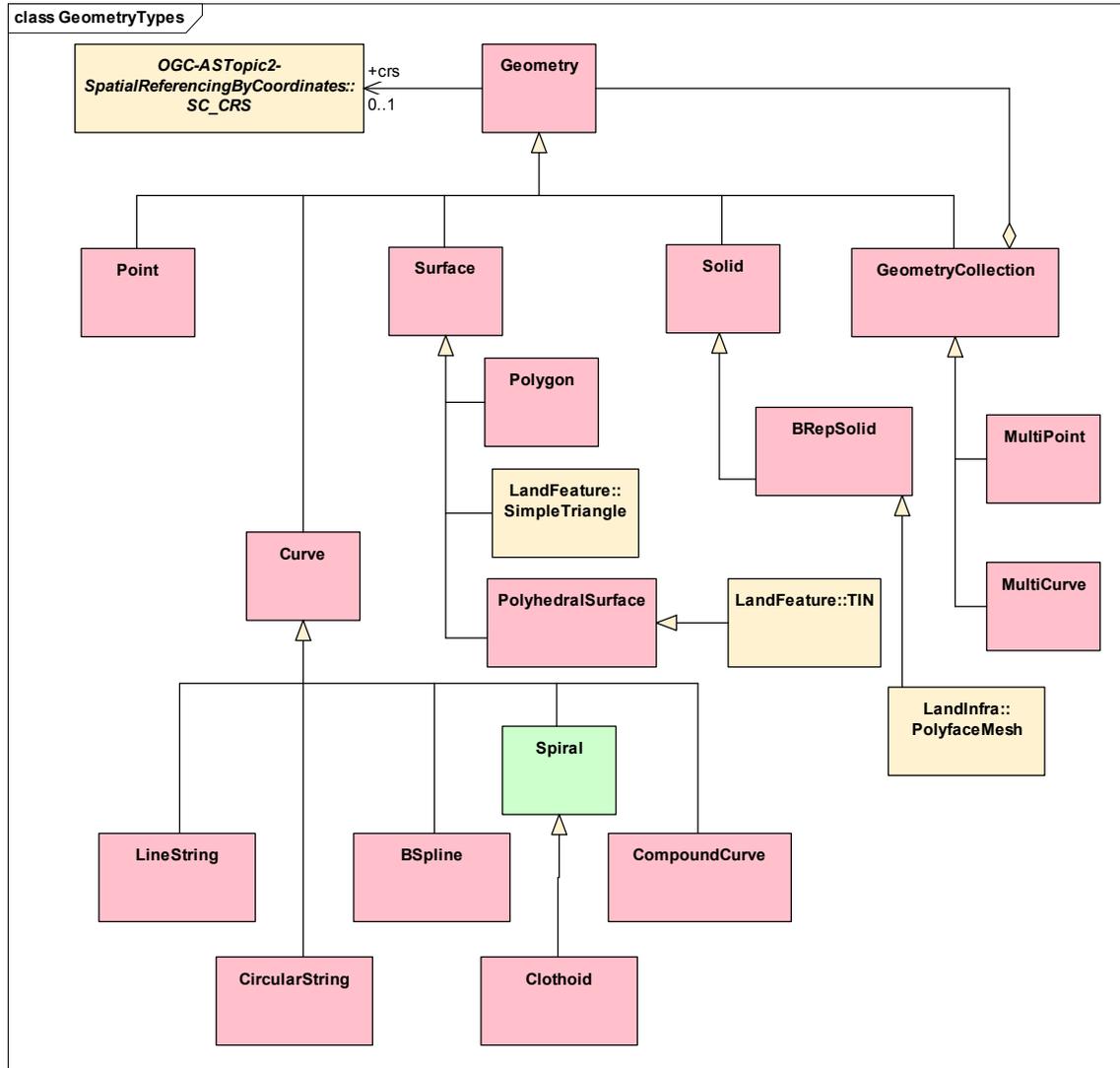


Figure 6. OGC Abstract Specification Topic 1 - Geometry Types with Extensions

Requirement	/req/land-infra/topic-1
	<p>A Land and Infrastructure encoding shall support the geometry types specified in the OGC Abstract Specification Topic 1, Feature Geometry as shown in pink and the SimpleTriangle, TIN, and PolyfaceMesh extensions shown in beige in Figure 6, that are appropriate to the supported subject area(s).</p> <p>Additional geometry types, not found in Topic 1, but required by a specific requirements class, are specified in that requirements class.</p>

Geometry types in pink in Figure 6 are from the OGC Abstract Specification Topic 1 – Geometry. The source for this is ISO 19107 - Geographic information — Spatial schema, which is currently under revision in ISO TC211. The GM\_ prefix has been dropped from the Class names in this figure because they will not appear in the revised 19107. Otherwise, the names are the same, except that GM\_Object changes to Geometry.

ISO 19107 allows that the CRS may be specified for an individual Geometry instance as long as all points in that Geometry have the same CRS. If specified, this overrides the LandInfraDataset.defaultCRS.

Geometry types in beige are not in Topic 1 and are therefore defined in this LandInfra specification in the most appropriate Requirements Class clause, as noted in the figure.

The Spiral type shown in green (as well as its Clothoid subtype) is in the current ISO 19107:2016 revision and is therefore anticipated to make its way into Topic 1 once the latter is consequently revised accordingly.

#### **7.2.5.1 Simple Triangle**

A SimpleTriangle is a surface defined by three points. This is a simplification of the Topic 1 Polygon, the latter requiring intermediary ring(s). See LandFeatures SimpleTriangle in 7.9.8.1.

#### **7.2.5.2 TIN**

A LandInfra TIN is an extension of the Topic 1 TIN which is really only useful if only the owned triangles are of interest and if the TIN is relatively small. The LandInfra TIN allows for the specification of various types of TIN Elements used in the construction of the TIN as well as the resultant triangles. However, these triangles are of the more compact SimpleTriangle type. See LandFeatures TIN in 7.9.8.

#### **7.2.5.3 Polyface Mesh**

A Polyface Mesh is a compact specification for the boundary of a Solid, comprising 3- (triad) and 4-sided (quad) faces. The faces are defined by three or four point indexes, respectively, where the point indexes refer to indexed (id'd) points.

## 7.2.6 OGC Abstract Specification Topic 19 – Linear Referencing

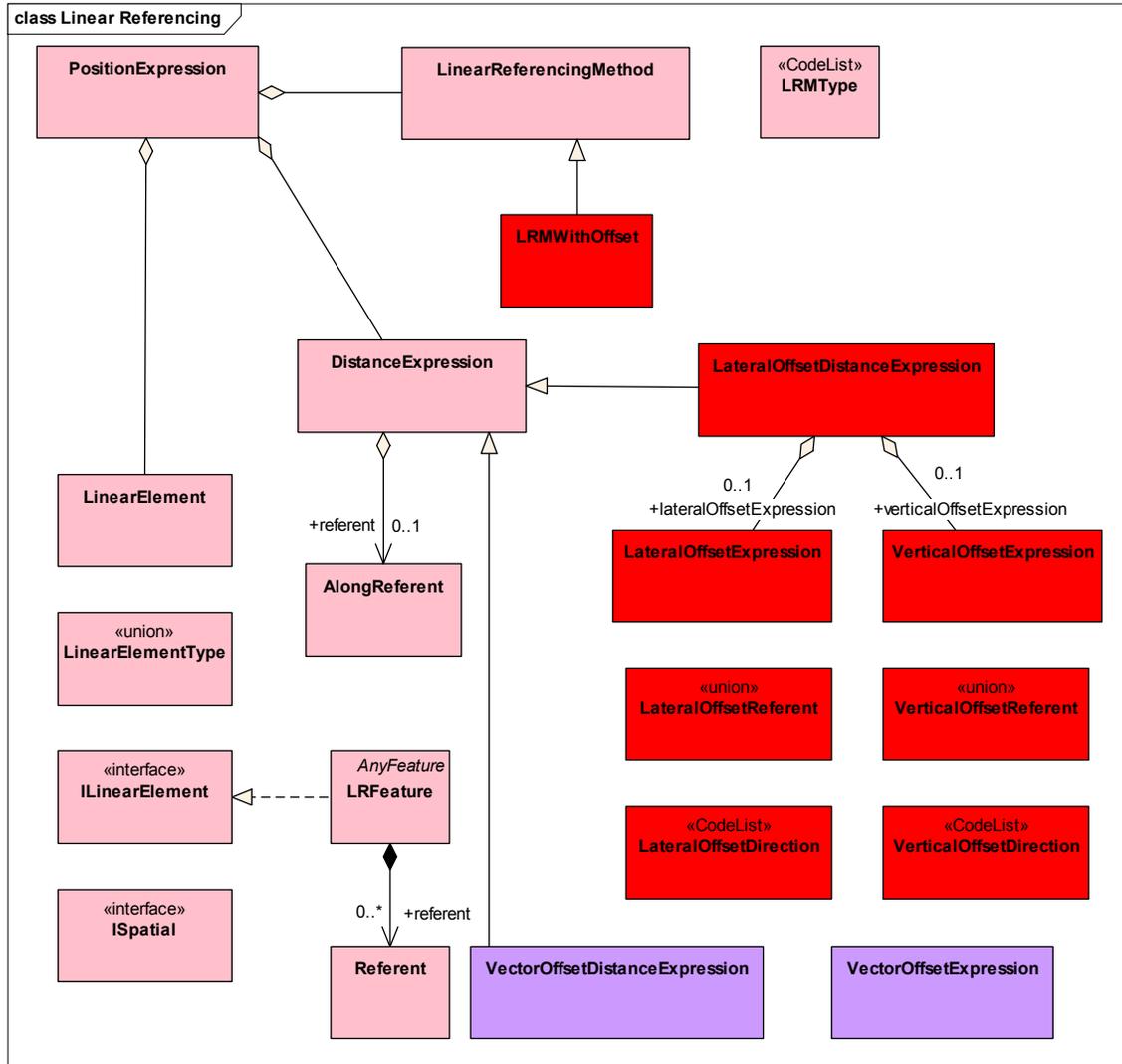


Figure 7. OGC Abstract Specification Topic 19 - Linear Referencing

All LandInfra Feature instances have an optional linearlyReferencedLocation attribute of type LinearlyReferencedLocation. This enables the location of the Feature to be specified by linear referencing, that is, as a distance along (and possibly offset from) some linear element.

Linear referencing has been standardized in ISO 19148 and adopted by OGC as Topic 19, Linear Referencing. Classes shown in pink in Figure 7 are required to satisfy the Topic 19 mandatory core Linear Referencing (LR) Package. In Topic 19, Class names are

prefixed with an abbreviation signifying their owning Package. These have been dropped from the Class names shown herein.

Topic 19 also includes several optional extension packages. Classes shown in red in Figure 7 are required to satisfy the optional Linear Referencing with Offsets (LRO) Package. Classes shown in purple in Figure 7 are required to satisfy the optional Linear Referencing with Vector Offsets (LROV) Package. Classes shown in orange in Figure 8 are required to satisfy the optional Linearly Located Event (LLE) Package.

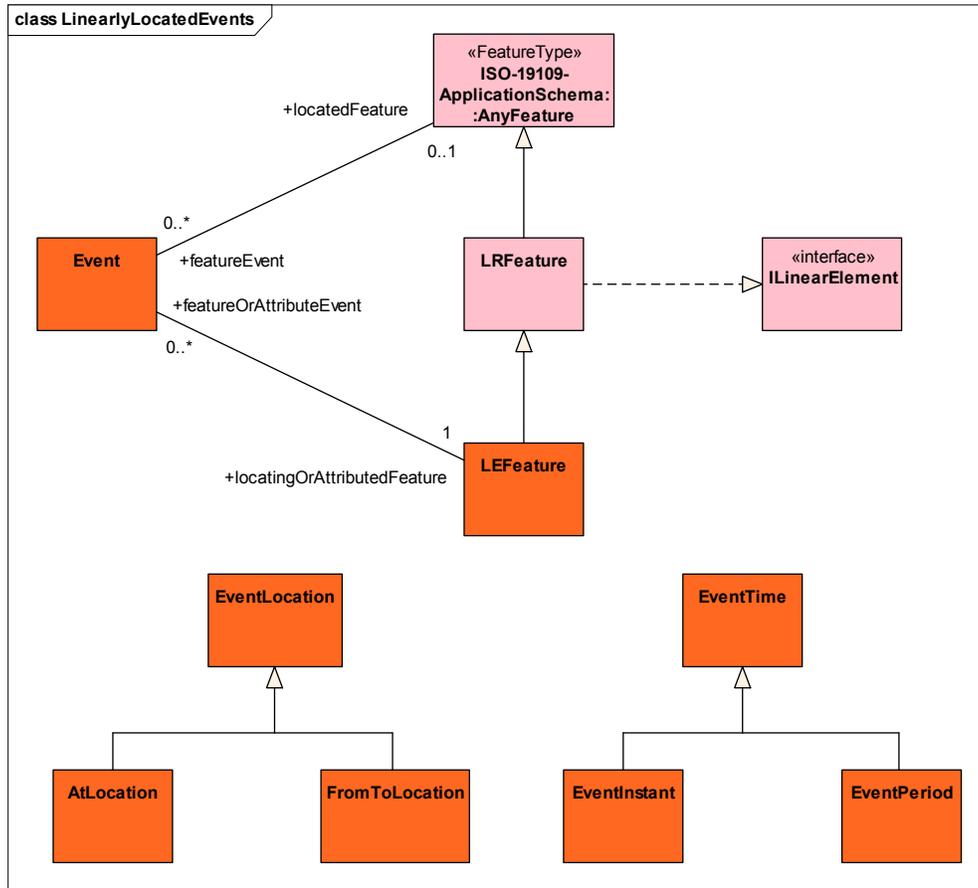


Figure 8. OGC Abstract Specification Topic 19 – Linearly Located Events

<b>Requirement</b>	<p><b><a href="#">/req/land-infra/topic-19-core</a></b></p> <p>If a Land and Infrastructure encoding supports linearly referenced locations, then the encoding shall support the PositionExpression and related Classes specified in the OGC Abstract Specification Topic 19, core Linear Referencing (LR) Package shown in pink in Figure 7 that are appropriate to the supported subject area(s).</p>
--------------------	---

<b>Requirement</b>	<b><a href="#">/req/land-infra/topic-19-extensions</a></b>  If a Land and Infrastructure encoding supports linearly referenced locations, then the encoding shall specify which of the OGC Abstract Specification Topic 19, Linear Referencing extension packages it supports and shall therefore support the appropriate Classes shown in Figure 7 and Figure 8 for those packages that are appropriate to the supported subject area(s).
--------------------	--

### 7.2.6.1 Direction

As specified in Topic 19, a linear element has a predefined direction from its start to its end. Distance along is measured in that direction. The offset terms “left”, “right”, “up”, and “down” are also relative to the direction of the linear element.

## 7.2.7 ISO 19109 Application Schema

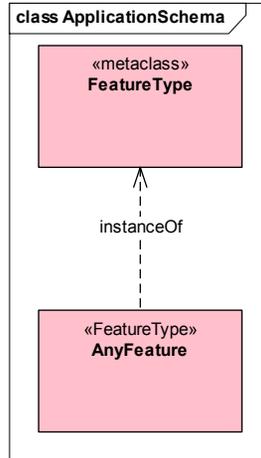


Figure 9. ISO 19109 - Rules for application schema – Feature Type

<b>Requirement</b>	<b>/req/land-infra/19109</b>  Features shall be supported by a Land and Infrastructure encoding in accordance with the uml:feature requirement in ISO 19109:2015 Clause 8.2.6 and shown in Figure 9, as follows:  Each instance of FeatureType shall be implemented by the encoding's equivalent of a UML Class having a generalization association with AnyFeature and with a stereotype of <<FeatureType>>.
--------------------	---

### 7.3 Requirements class: Facility

Requirements Class	
<b>/req/facility</b>	
Target type	Encoding of the conceptual model
Name	Facility
Dependency	<b>/req/land-infra</b>
Requirement	<b>/req/facility/classes</b>
Requirement	<b>/req/facility/subtypes</b>

Facilities include buildings and civil engineering works and their associated siteworks. Civil engineering works, or infrastructure facilities, are construction works comprising a structure, such as a dam, bridge, road, railway, runway, utilities, pipeline, or wastewater system, or are the result of operations such as dredging, earthwork, and geotechnical processes.

A facility has a life cycle, including planning, design, construction, maintenance, operation, and removal phases. Design and construction phases are typically performed as part of a project. There may be multiple such projects during the life cycle of the facility to enable phased construction and incremental improvement.

buildingSMART International (bSI) has traditionally been mostly focused on the design/construct aspect of a facility within the context of an individual improvement project. OGC takes a broader view of the full life cycle of the facility which may include one or more design/construction projects for phased construction or incremental improvement over time. The OGC Conceptual Model therefore includes classes for Facility and its various sub-types, though these are not very highly attributed in this Land and Infrastructure standard.

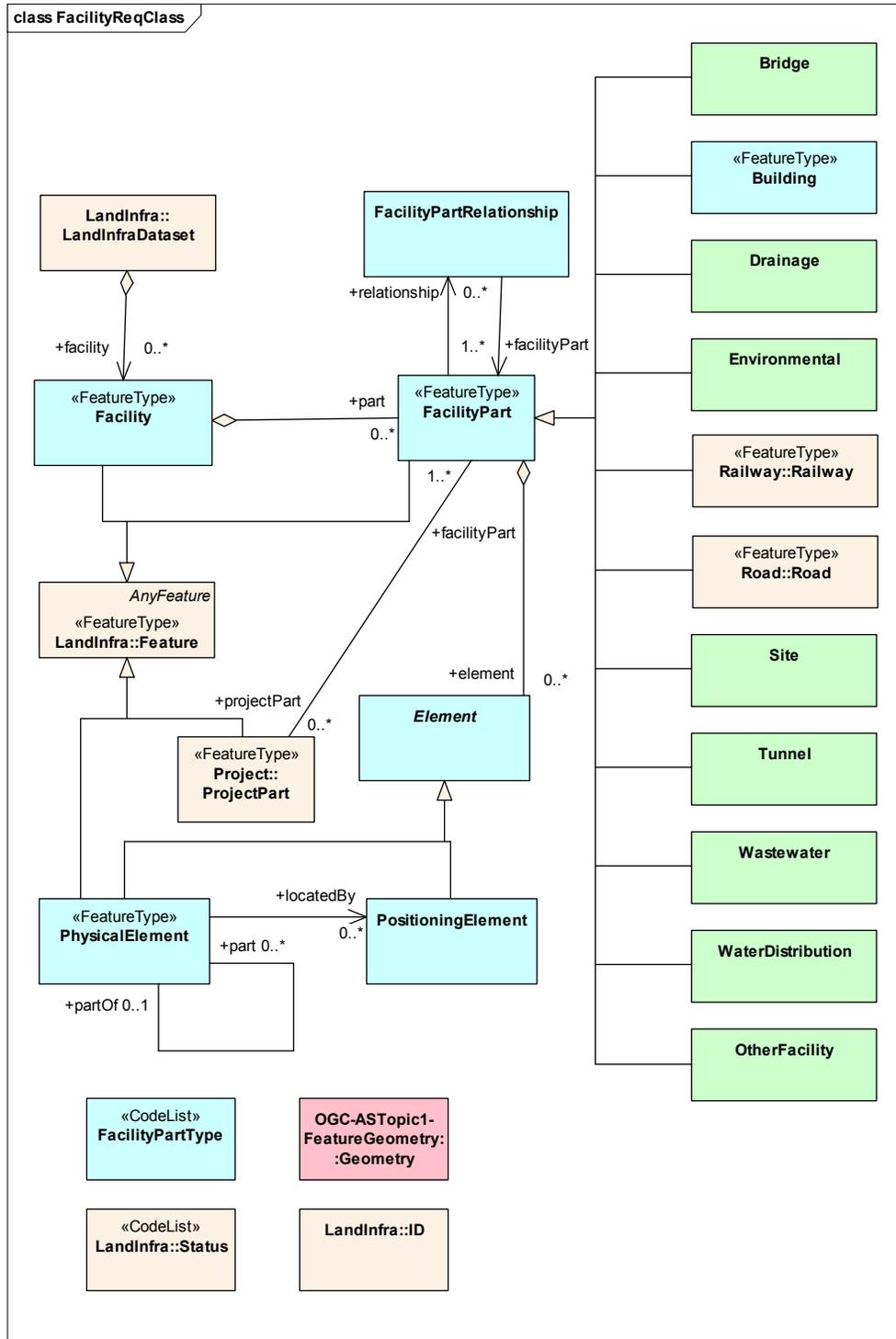


Figure 10. Facility Requirements Class



**status:** Facility's status

**footprint:** optional geometric extent of the Facility

**part:** any number of FacilityParts

### 7.3.1.2 Facility Part

Facilities may be broken down into FacilityParts, based on the type of facility. For example, a shopping mall may include buildings, roads, site, drainage, water distribution and wastewater. A road facility (e.g., a motorway) may be decomposed into multiple roads, where one of these roads is the non-branching mainline between two specified junctions. Further restrictions on partitioning facilities may be dictated by individual part types. For example, a road part must be a single alternative for a continuous, non-branching, and non-overlapping section of road.

FacilityParts are roughly equivalent to IfcSpatialElement (e.g., building, road). These provide containment for IfcElements (e.g., windows, curb and gutter).

As a subtype of Feature, FacilityPart inherits attributes featureID, name, description, spatialRepresentation, linearlyReferencedLocation, property, and propertySet. FacilityParts have the following additional attributes:

**facilityPartID:** user defined ID used to identify a facility part, unique within a Facility or globally unique with the inclusion of an ID.scope value

**type:** one of the code listed FacilityPartTypes

**status:** FacilityPart status

**footprint:** optional geometric extent of the FacilityPart

**alternative:** optional design alternative designation

**projectPart:** optional part(s) of a project responsible for the creation or removal of this FacilityPart

**relationship:** optional relationship between this FacilityPart and another FacilityPart

### 7.3.1.3 Facility Part Type

Within the Facility Requirements Class, FacilityParts are classified by FacilityPartType code list values. A more detailed specification would be provided by FacilityPart subtypes in their own respective Requirements Classes (see 7.3.3, Facility Part Subtypes).

The FacilityPartType values are as follows, with most definitions coming from ISO 6707-1:2014:

**bridge:** civil engineering works that affords passage to pedestrians, animals, vehicles, and services above obstacles or between two points at a height above ground

**building:** construction works that has the provision of shelter for its occupants or contents as one of its main purposes, usually partially or totally enclosed and designed to stand permanently in one place

**drainage:** system of drains (pipe, channel, or tunnel) and ancillary works (inlets, outlets, manholes) that conveys surface water to an outfall or other place of disposal

**environmental:** civil engineering works whose primary function is environmentally related, such as a fish run

**railway:** continuous, non-branching, and non-overlapping section of a national or regional transport system for guided passage of wheeled vehicles on rails

**road:** continuous, non-branching, and non-overlapping way, mainly for vehicles

**site:** construction works or landscape work on land associated with, and adjacent to, civil engineering works or a building

**tunnel:** horizontal or sloping underground or underwater enclosed way of some length

**wastewater:** system of sewers and ancillary works that conveys the contents to a sewage treatment works or other place of disposal. Wastewater (or sewerage, or sewage) is water discharged after being used in a household or in a process, or produced by a process, other waters in a combined system and water that has infiltrated a sewerage system.

**waterDistribution:** water supply system, including pipes, valves, and pumps

**other:** any facility part type not covered by the explicit types above

#### 7.3.1.4 Facility Part Relationship

Facility parts may be related to each other, such as a road connecting into a roundabout or a road being supported by a bridge.

**relationship:** the relationship between one part and another (e.g., “connected-to”, “supported-by”)

**description:** supplemental information describing the relationship

**facilityPart:** the FacilityPart that this FacilityPart is related to

### 7.3.1.5 Element

FacilityPart is a logical collection of Elements (see Figure 12). Elements can represent physical (manufactured or constructed) parts of the FacilityPart (PhysicalElement). Alternatively, they can be virtual elements used to locate, align, or organize other elements (PositioningElement). Note: for bSI, Elements are instead owned by ProjectParts, because of their project (vs. facility) focus.

Element is an abstract class. Instantiable Element subtypes specify physical and positioning elements. Element has the following attributes:

**elementID:** user defined ID used to identify an element, unique within the LandInfra dataset or globally unique with the inclusion of an ID.scope value

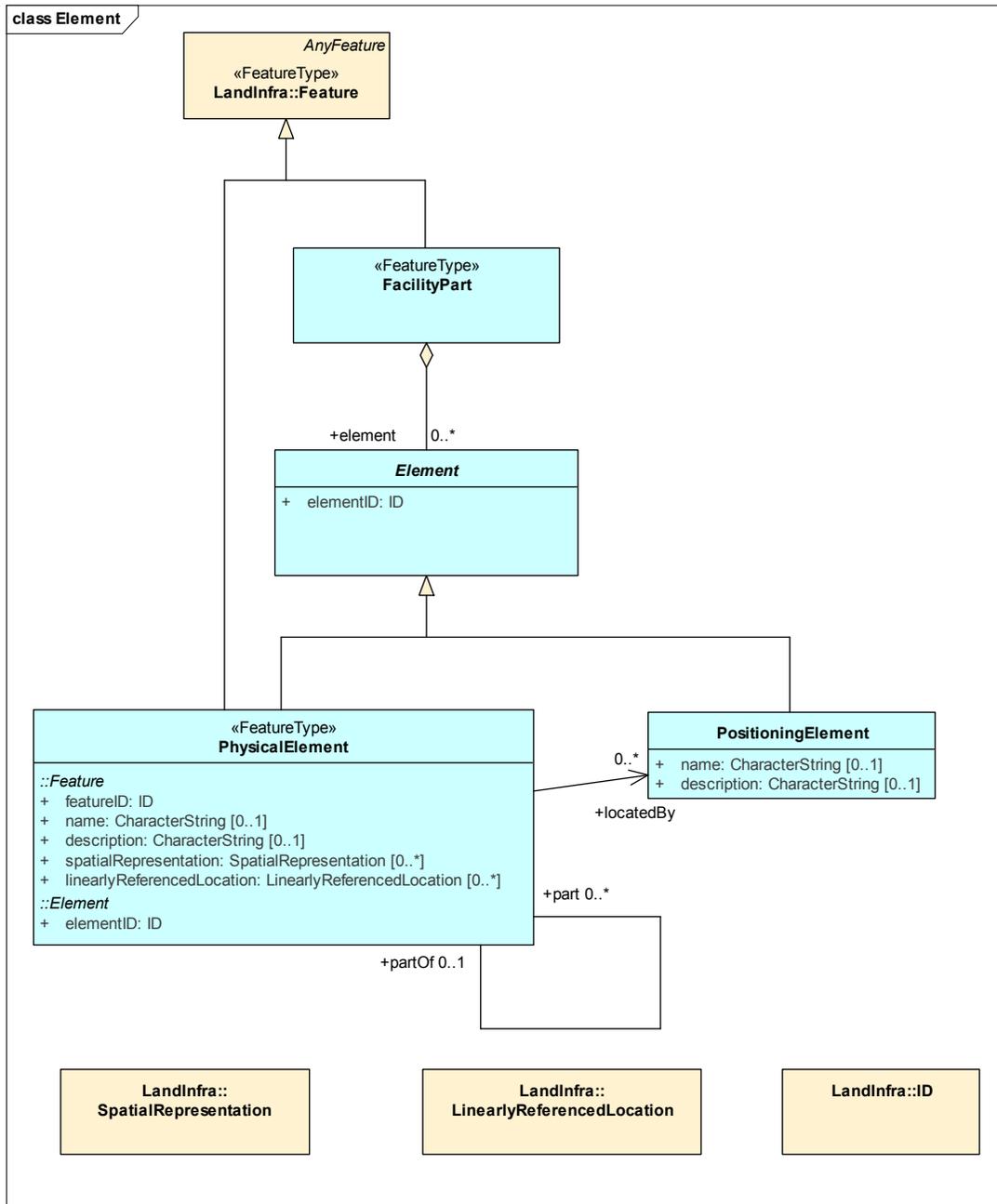


Figure 12. Element Context Diagram

### 7.3.1.6 Physical Element

PhysicalElement specifies a physical part of a FacilityPart. As a type of Feature, it inherits attributes featureID, name, description, spatialRepresentation, linearlyReferencedLocation, property, and propertySet. As a type of Element, it inherits elementID.

PhysicalElement has the following additional attributes:

**part:** if the Element is made up of other Elements, then these are those part Elements

**part of:** if the Element is a part of another Element, this is the Element of which it is a part

**locatedBy:** optional PositioningElement(s) used to locate this PhysicalElement

For bSI, IfcElement has subtypes including IfcBuildingElement, IfcElementComponent, and IfcElementAssembly:

IfcBuildingElement comprises all elements that are primarily part of the construction of a building, i.e., its structural and space separating system.

IfcElementComponent is a representation for minor items included in, added to or connecting to or between elements, which usually are not of interest from the overall building structure viewpoint. However, these small parts may have vital and load carrying functions within the construction. These items do not provide any actual space boundaries. Typical examples of IfcElementComponents include different kinds of fasteners and various accessories.

IfcElementAssembly represents complex element assemblies aggregated from several elements, such as discrete elements, building elements, or other elements.

EXAMPLE: Steel construction assemblies, such as trusses and different kinds of frames, can be represented by the IfcElementAssembly entity. Other examples include slab fields aggregated from a number of precast concrete slabs or reinforcement units made from several reinforcement bars. Also bathroom units, staircase sections and other premanufactured or precast elements are examples of the general IfcElementAssembly entity.

The IfcElementAssembly shall represent an aggregate, i.e. it should have other elements, being subtypes of IfcElement, as contained (sub)parts.

The IfcElementAssembly is an aggregate i.e. being composed by other elements and acting as an assembly using the objectified relationship IfcRelAggregates, referring to it by its inverse attribute SELF\IfcObjectDefinition.IsDecomposedBy. Components of an assembly are described by instances of subtypes of IfcElement.

In this case, the contained subtypes of IfcElement shall not be additionally contained in the project spatial hierarchy, i.e. the inverse attribute SELF\IfcElement.ContainedInStructure of those IfcElement's shall be NIL.

As a subtype of IfcObject, IfcElement derives from IfcObjectDefinition and therefore has the IsDecomposedBy attribute and is able to get aggregated in multiple levels.

In this LandInfra Standard, decomposition of PhysicalElement to any number of levels is permitted and optional. Beyond this, the requirement for aggregate or component classes has not yet been identified.

Additional attributes are defined for specific PhysicalElement subtypes.

### 7.3.1.7 Positioning Element

PositioningElement is used to specify where a PhysicalElement is located. This might include a grid for building PhysicalElements or an Alignment for linear infrastructure (e.g., Road) PhysicalElements. As a type of Element, it inherits elementID.

PositioningElement has the following additional attributes:

**name:** optional name of the PositioningElement

**description:** optional description of the PositioningElement

Additional attributes are defined for specific PositioningElement subtypes.

### 7.3.2 Facility Requirements Class Associated Classes

The Facility Requirements Class is dependent upon the core LandInfra Requirements Class. The Project, Alignment, Road, and Railway Requirements Classes are directly dependent upon the Facility Requirements Class.

### 7.3.3 Facility Part Subtypes

<b>Requirement</b>	<b><a href="#">/req/facility/subtypes</a></b> A Land and Infrastructure encoding shall specify which of the FacilityPart subtypes shown in Figure 10 it supports.
--------------------	--

Road and Railway are included in the initial release of this Land and Infrastructure Conceptual Model Standard. Others may be added as future extensions.

## 7.4 Requirements class: Project

Requirements Class	
<b>/req/project</b>	
Target type	Encoding of the conceptual model
Name	Project
Dependency	<b>/req/facility</b>
Requirement	<b>/req/project/classes</b>
	The Project Requirements Class Classes shown in blue in Figure 13 shall be provided for by the encoding in a manner consistent with the encoding.

A project is an activity related to the improvement of a facility, including design and/or construction. A project may be for the creation, modification, or elimination of the entire facility or a part of the facility. A LandInfra dataset may contain any number of Projects, though typically a LandInfra dataset will have at most one project.

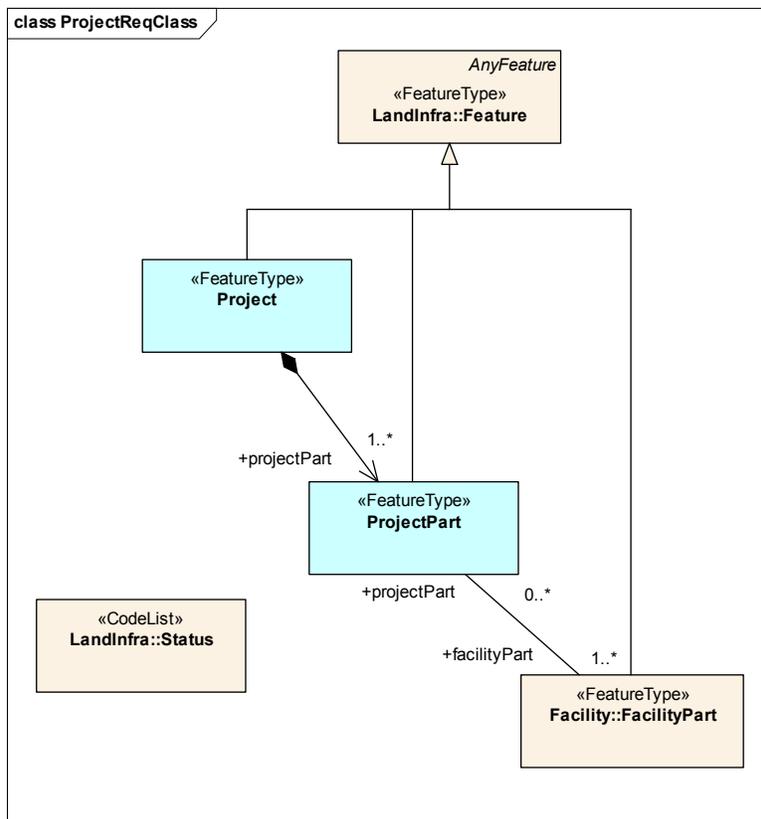


Figure 13. Project Requirements Class

## 7.4.1 Project Requirements Class Classes

### 7.4.1.1 Project

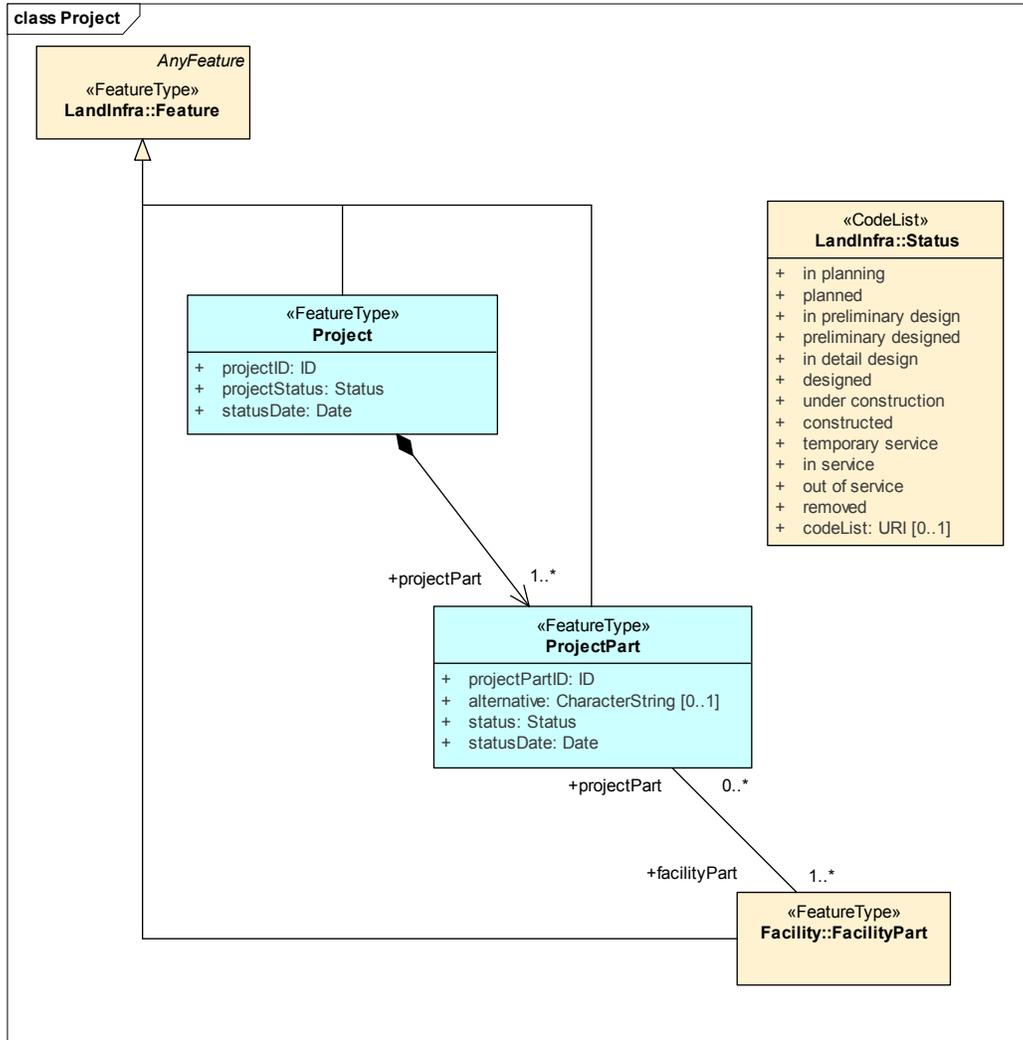


Figure 14. Project and ProjectPart Context Diagram

As a subtype of Feature, Project inherits attributes featureID, name, description, spatialRepresentation, linearlyReferencedLocation, property, and propertySet. Project has the following additional attributes:

**projectID:** user defined identifier for the Project, unique within a LandInfra dataset or globally unique with the inclusion of an ID.scope value

**projectStatus:** status of the Project

**statusDate:** date on which the status value was valid

**projectPart:** one or more ProjectParts

#### 7.4.1.2 Project Part

Projects can have multiple parts, termed ProjectParts. Such parts relate to a single facility type (road, railway, etc.). It may be necessary to further break these parts down based on some set of constraints, for example, design alternative or construction phase. This keeps the ProjectParts simple and more manageable.

As a subtype of Feature, ProjectPart inherits attributes featureID, name, description, spatialRepresentation, linearlyReferencedLocation, property, and propertySet. ProjectPart has the following additional attributes:

**projectPartID:** user defined identifier for the ProjectPart, unique within a Project or globally unique with the inclusion of an ID.scope value

**alternative:** optional design or construction alternative designation (e.g., bypass alternative, temporary construction re-alignment)

**status:** status of the ProjectPart

**statusDate:** date on which the status value was valid

**facilityPart:** that part(s) of the Facility for which this ProjectPart applies

#### 7.4.2 Project Requirements Class Associated Classes

The Project Requirements Class is dependent upon the core LandInfra Requirements Class and the Facility Requirements Class.

## 7.5 Requirements class: Alignment

Requirements Class	
<b>/req/alignment</b>	
Target type	Encoding of the conceptual model
Name	Alignment
Dependency	<b>/req/facility</b>
Requirement	<b>/req/alignment/classes</b>
Requirement	<b>/req/alignment/description</b>
Requirement	<b>/req/alignment/CRS</b>
Requirement	<b>/req/alignment/LR-interfaces</b>
Requirement	<b>/req/alignment/measures</b>

Requirement	<p><b>/req/alignment/classes</b></p> <p>The Alignment Requirements class Classes shown in blue in Figure 15 shall be provided for by any encoding which includes Alignments and then in such a manner as is consistent with the encoding.</p>
-------------	---

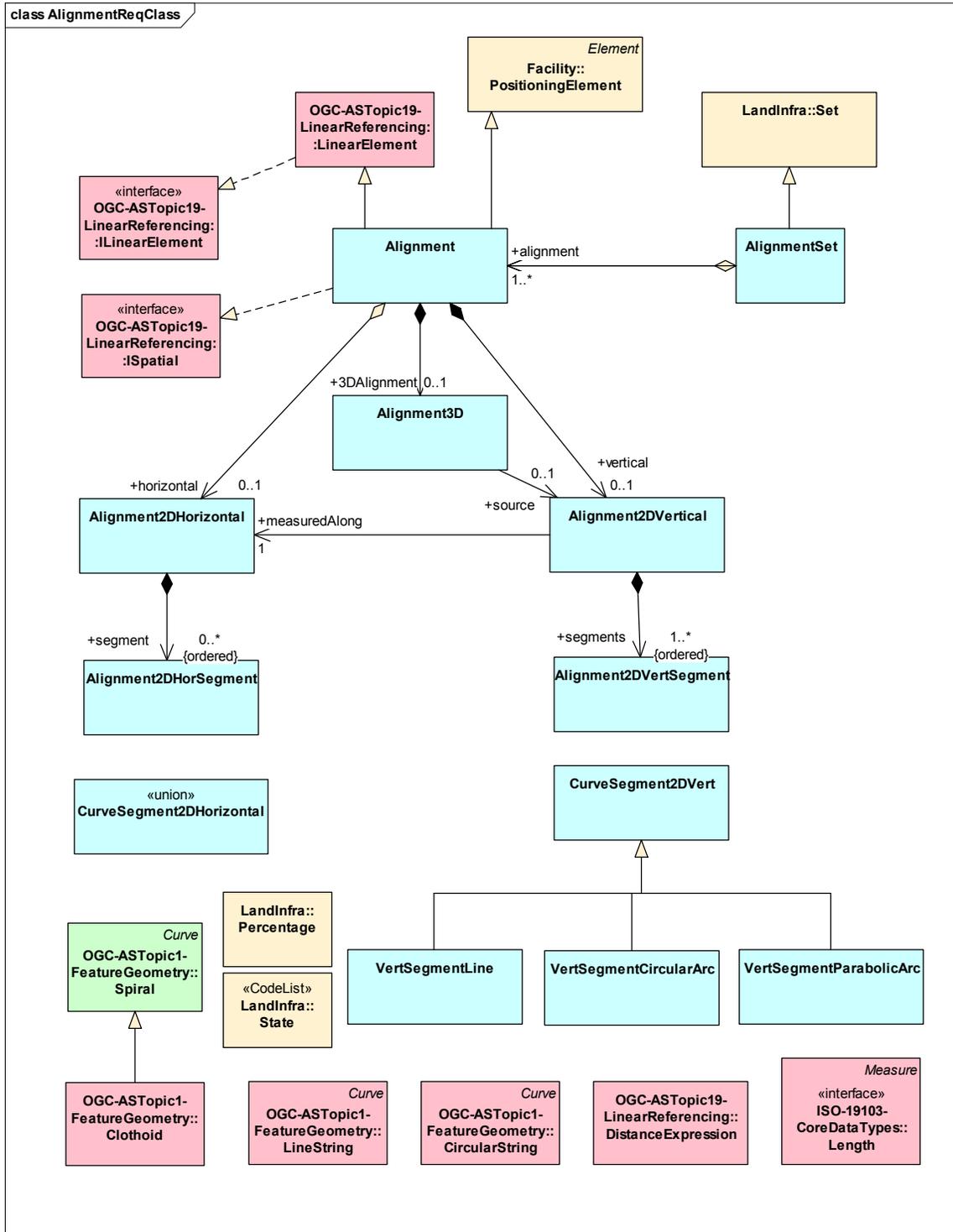


Figure 15. Alignment Requirements Class



The context diagram for Alignment is shown in Figure 16.

An Alignment shall be continuous, non-overlapping, and non-branching (though it may contain intersections with other Alignments). If it is used within the context of a Project, it shall be for a single alternative, as specified by the ProjectPart.

Requirement	/req/alignment/description
	An encoding which includes Alignments shall specify that all of the included Alignments shall be continuous, non-overlapping, and non-branching (though it may contain intersections with other Alignments) and that if it is used within the context of a Project, the included Alignment shall be for a single alternative, as specified by the ProjectPart.

FacilityParts such as Roads may have any number of Alignments. Typically, there will be an Alignment for the centerline of a road. For dual carriageway roads, each carriageway may be a separate Road FacilityPart. Each could therefore have a separate Alignment, allowing them to have differing geometries. However, they might also share a reference horizontal alignment which is defined along the approximate centerline of the entire road. To distinguish where along the Road the Alignment is defined, a name and description would be specified.

Locations along the Alignment can be specified as linearly referenced locations based upon some Linear Referencing Method (LRM). Traditionally the LRM has been limited to stationing, but there is now support for easing this constraint so as to allow for other LRMs. Because PhysicalElements will be located along the Alignment using linearly referenced locations, a default Linear Referencing Method (LRM) for the Alignment must be specified. This will also be the LRM used by the Alignment2DVertical to specify distances along the Alignment2DHorizontal. See further Linear Referencing discussion below.

In addition to the elementID attribute inherited from Element and the name and description attributes inherited from PositioningElement, Alignment has the following additional attributes:

**lrm:** the default Linear Referencing Method for the Alignment and the LRM used for the HorizontalAlignment distanceAlong values specified in the VerticalAlignment geometries.

**horizontal:** optional Alignment2DHorizontal value. Mandatory if the Alignment is defined in 2D

**vertical:** optional Alignment2DVertical value. Only applicable if Alignment is defined in 2D and if an Alignment2DHorizontal value is also defined.

**3DAlignment:** optional Alignment3D value. Only required if the Alignment is defined in 3D.

### 7.5.2 Alignment Set

An AlignmentSet is a set of Alignments grouped together for a particular reason. As a subtype of Set, it inherits the Set attributes of name, description, and authority. AlignmentSet has the following additional attribute:

**alignment:** one or more Alignments

### 7.5.3 Horizontal Alignment

An Alignment2DHorizontal is a geometry/location projected onto the horizontal plane. If defined by a 2DLineStringRepresentation, the points of this LineString have two coordinate values, based upon the applicable 2D Coordinate Reference System (CRS). If defined by a list of Alignment2DHorSegments, the geometry/location of these segments shall be in a Cartesian local engineering reference system specified using the Alignment2DHorizontal.crs. The point coordinates are typically x and y but might alternatively be defined as Northing and Easting values.

<b>Requirement</b>	<b>/req/alignment/CRS</b>  All geometry/locations for Alignment2DHorSegments and Alignment2DVertSegments shall be specified using the Alignment2DHorizontal.crs which shall be a Cartesian local engineering reference system.
--------------------	--

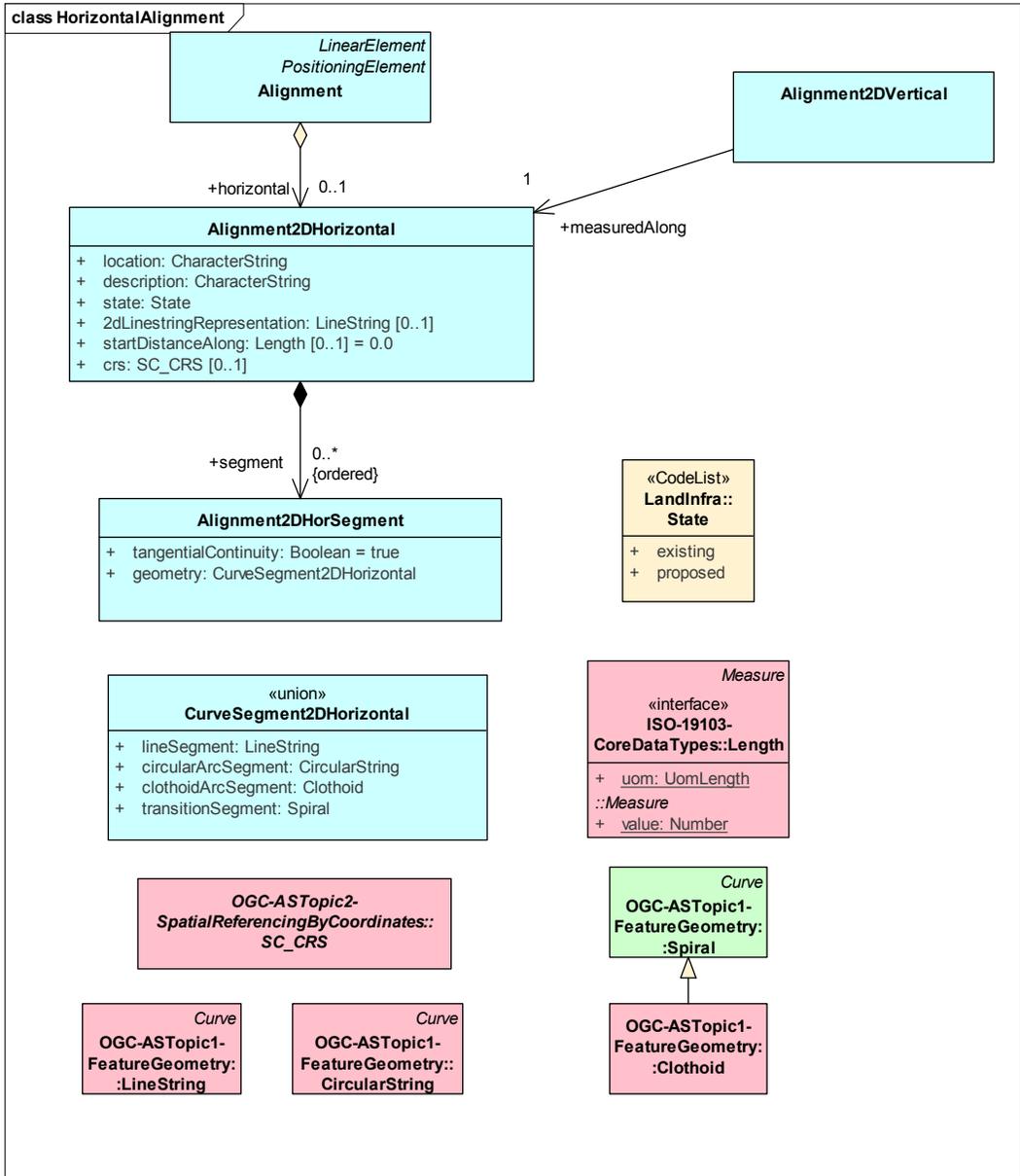


Figure 17. Horizontal Alignment

An Alignment may be defined solely by an Alignment3D, making Alignment2DHorizontal an optional part of Alignment.

The simplest way to define the geometry/location of an Alignment2DHorizontal is with a two-dimensional LineString. The CRS is either the defaultCRS of the LandInfraDataset or the overriding CRS of the LineString. This 2dLinestringRepresentation is more apt to be supportable by a full range of software, including sketching tools, GIS, CAD, and Civil Design Applications.

A more detailed representation suitable for construction purposes can be defined using Alignment2DHorSegments in a Cartesian engineering reference system. Due to the complexity of the curve types used, Alignment2DHorSegments may only be supported by CAD and Civil Design Application software.

A more detailed geometric representation of the Alignment2DHorizontal geometry is developed during preliminary and detailed design. Straight lines define the rough geometry and then circular curves having constant radius can be added to smooth out the alignment. Transition curves in the form of clothoids or other spirals having increasing or decreasing radii can be introduced in between the lines and circular curves (or between pairs of circular curves) to smooth out the transitions. At any time during this design process, the straight line, circular curve, clothoid, or other spirals are represented as individual, connected segments called Alignment2DHorSegments.

Because Alignment2DHorSegments are connected, the end of the previous segment should match the start location of the next segment. There is no requirement that the subsequent segment be tangentially continuous with the preceding one. Roadway design practice allows for discontinuities of small deflection angles. However, this would mean that some implementation convention would need to be stated addressing how templates or cross sections would be applied at this location. Other improvements which use alignments, such as retaining walls or transmission lines would not have a continuity constraint. The Boolean-valued “tangentialContinuity” (default = true) indicates that the current segment is continuous with the previous one.

The OGC Abstract Specification Topic 1, Geometry, already provides an acceptable concept model for geometries such as LineString, CircularArc, and Clothoids. These are consequently included in LandInfra by reference to and dependence upon this OGC Abstract Specification. The proposed ISO 19107:2016 revision adds support for Spirals.

The conceptual model for Horizontal Alignment is shown in Figure 17.

Alignment2DHorizontal has the following attributes:

**location:** the location of the Alignment, e.g., "road centerline"

**description:** describes where the alignment is located with respect to the owning road or railway segment, such as "road centerline"

**state:** existing or proposed

**2dLinestringRepresentation:** optional SpatialRepresentation as a two-dimensional LineString

**startDistanceAlong:** if the Alignment LRM is of type "absolute", then the measured value at the start of the horizontal alignment, if other than 0

**crs:** if defined by segments, the Cartesian engineering reference system for the segment geometries

**segment:** zero or more, ordered, Alignment2DHorSegment values

### 7.5.3.1 Alignment2DHorSegment

An Alignment2DHorizontal can be specified as a series of ordered Alignment2DHorSegments, distinguished by their geometry.

Alignment2DHorSegments have the following attributes:

**tangentialContinuity:** True (default) if the segment is tangentially continuous with the previous segment

**geometry:** the geometry of the segment, defined by CurveSegment2DHorizontal

### 7.5.3.2 CurveSegment2DHorizontal

CurveSegment2DHorizontal provides the choice of geometry for an Alignment2DHorSegment:

**lineSegment:** the segment geometry is of type LineString

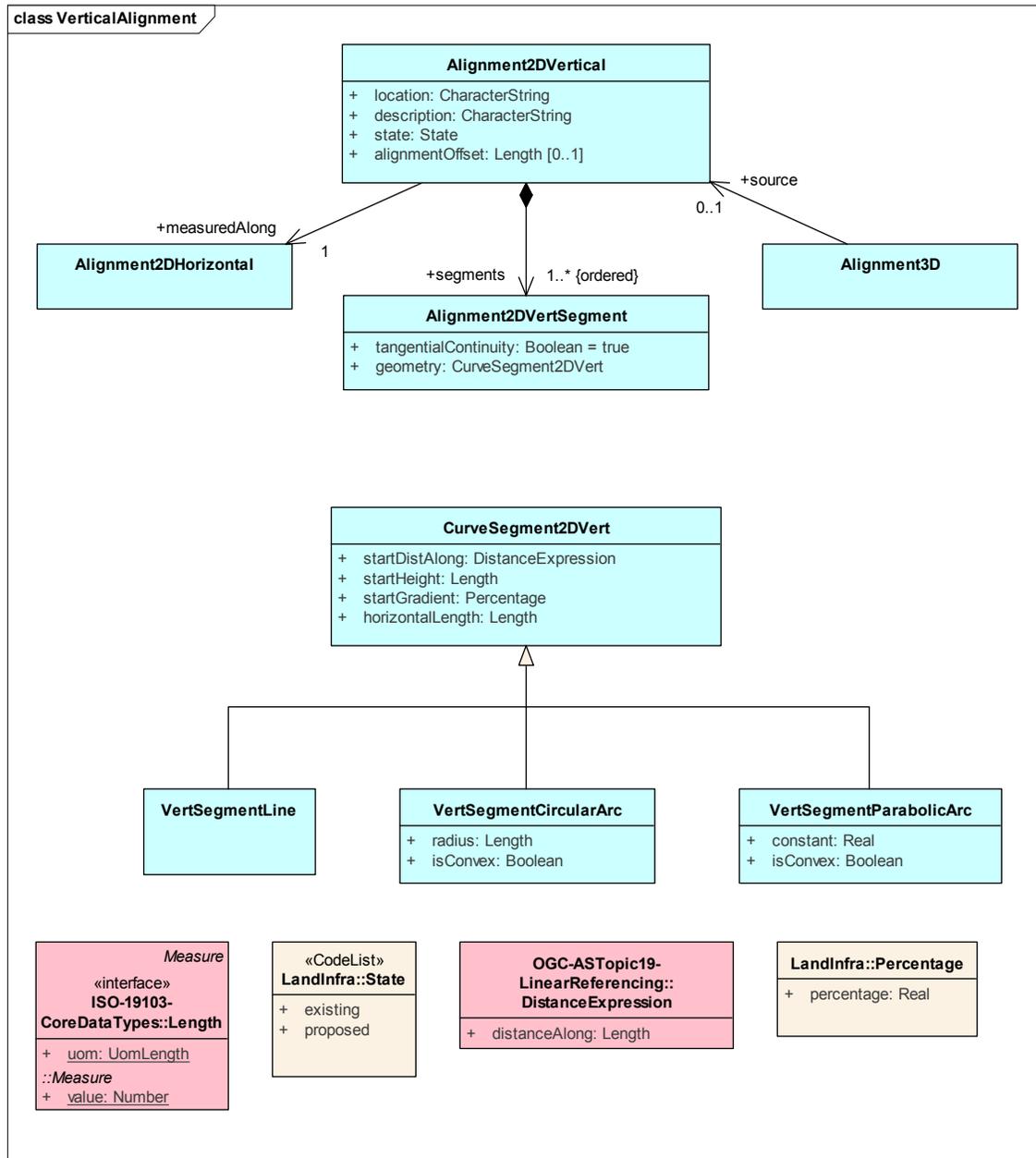
**circularArcSegment:** the segment geometry is of type CircularArc

**clothoidArcSegment:** the segment geometry is of type Clothoid

**transitionSegment:** the segment geometry is of the proposed ISO 19107 Spiral type

## 7.5.4 Vertical Alignment

An Alignment2DVertical specifies a geometry/location in the vertical direction for the Alignment2DHorizontal using the Cartesian engineering reference system specified by the Alignment2DHorizontal crs value. Imagine a vertical surface generated by following along a Alignment2DHorizontal. This is often referred to as a long- (vs. cross-) section. If this surface is flattened into a vertical plane, the horizontal dimension is the distance along the Alignment2DHorizontal. The other, perpendicular axis is used to specify the elevation value.



### *Figure 18. Vertical Alignment*

In order for an Alignment to have an Alignment2DVertical, it must also have a single Alignment2DHorizontal defined by Alignment2DHorSegments. Together these segments serve as the linear element being measured along for defining the Alignment2DVertical.

The first coordinate value of an Alignment2DVertical point measures the distance along the Alignment2DHorizontal segments, using linearly referenced locations. These are dependent upon the Linear Reference Method specified for the Alignment. The second coordinate value represents the elevation value of the Alignment2DVertical point. As is the case with the x and y coordinates in the Alignment2DHorizontal, these are dependent upon the Cartesian engineering reference system.

To simplify things, this standard assumes that the LRM to be used for all of the Alignment2DVertical distance along measures shall be equal to the default LRM of the Alignment. As the bSI IFC Alignment project team has suggested that LRMs be limited to the “absolute” type, this leaves just the measured value for specifying distance along locations. Since measurements are along the Alignment2DHorizontal segments linear element, which is a projection onto the x/y plane, then the distance along measurements are also in the x/y projection and are not true on-the-elevated-road-surface measurements.

A single Alignment is limited to at most one Alignment2DHorizontal and one associated Alignment2DVertical. It may be necessary however for multiple vertical alignments to be defined for the same horizontal alignment, as in the following examples:

- 1 - two carriageway centerline profiles where the carriageways share a single horizontal alignment (in this case it may be necessary to define a horizontal offset measure for each)
- 2 - partial vertical profiles, as required in the use cases

In order to allow multiple vertical alignments to reference the same horizontal alignment, it is necessary to create new Alignments. Each of these Alignments can use the same Alignment2DHorizontal.

Alignment2DVertical has the following attributes:

**location:** short name for where the Alignment2DVertical exists, e.g. “centerline” or “edge of pavement”

**description:** further description of the Alignment2DVertical

**state:** existing or proposed

**alignmentOffset:** distance from the Alignment2DHorizontal segments to the Alignment2DVertical; not required if this distance is zero

**measuredAlong:** the Alignment2DHorizontal along which the distanceAlong measurements of the Alignment2DVertical are made

**segments:** one or more ordered segments which define the geometry of the Alignment2DVertical

#### 7.5.4.1 Alignment2DVertSegment

For consistency with the Alignment2DHorizontal, the Alignment2DVertical is defined as an ordered list of segments, called Alignment2DVertSegments. As was the case with Alignment2DHorSegments, tangential continuity is not mandatory. The Boolean-valued “tangentialContinuity” (default = true) indicates that the current segment is continuous with the previous one.

Alignment2DVertSegment has the following attributes:

**tangentialContinuity:** True (default) if the segment is tangentially continuous with the previous segment

**geometry:** the geometry of the segment, defined by CurveSegment2DVert

#### 7.5.4.2 CurveSegment2DVert

CurveSegment2DVert provides the choice of geometry for an Alignment2DVertSegment. It has the following attributes:

**startDistanceAlong:** the start linearly referenced location of the segment, specified as a distance along the horizontal alignment linear element in the 2D x, y plane, using the alignment’s LRM

**startHeight:** elevation (z coordinate) at the start of the segment

**startGradient:** the grade at the start of the segment specified as a positive or negative percentage, where a value of 2.0 (2%) indicates a rise of 2 in elevation for every 100 in distanceAlong

**horizontalLength:** the segment’s total length in the 2D x, y plane

The geometry of an Alignment2DVertSegment may be either:

- 1 - a linear segment of constant gradient (VertSegmentLine)
- 2 - a circular arc (VertSegmentCircularArc)

### 3 - a parabolic arc (VertSegmentParabolicArc)

These are differentiated as subtypes of CurveSegment2DVert. VertSegmentLine has no additional attributes. In addition to the attributes for CurveSegment2DVert, VertSegmentCircularArc has the following attributes:

**radius:** specifies the curvature of the VertSegmentCircularArc

**isConvex:** true or false

In addition to the attributes for CurveSegment2DVert, VertSegmentParabolicArc has the following attributes:

**constant:** parabola constant specifying the “steepness” of the parabola =  $K * 100$

**isConvex:** true or false

Both segment types have an isConvex Boolean attribute. A value of “true” (convex) means that the gradient at the beginning of the segment is less than the gradient at the end of the previous segment. A value of “false” (concave) (Boolean=“false”) means that the gradient at the beginning of the segment is greater than the gradient at the end of the previous segment.

Note: asymmetric parabolic curves are not included as a separate geometry type as they can instead be specified as two symmetric, contiguous, and tangentially continuous parabolic arcs.

### 7.5.5 3D Alignment

Once an Alignment2DVertical and its referenced Alignment2DHorizontal have been specified, it may be appropriate to combine the two 2-D geometric representations into a single 3-D representation called an Alignment3D. This resultant geometry is typically represented as a 3-D linestring of points, each having x, y, and z coordinate values or whatever is appropriate in the CRS of the linestring.

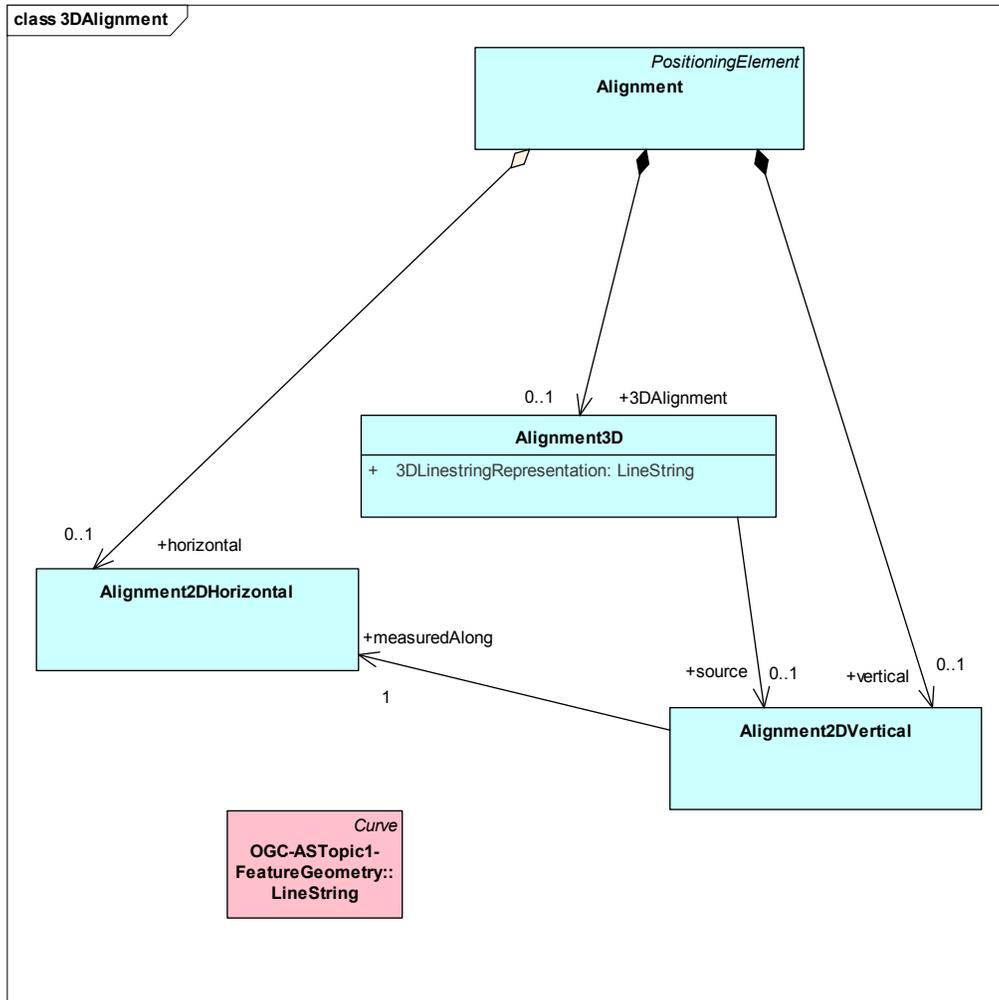


Figure 19. 3D Alignment

Alignment3D has the following attributes:

**3DLinestringRepresentation:** geometry of the Alignment3D specified as a LineString geometry

**source:** optional Alignment2DVertical (along with its measuredAlong Alignment2DHorizontal) whose combined geometry are used to calculate the geometry of the Alignment3D, but only if the Alignment3D is derived from these

It is also possible for an Alignment to have an Alignment3D without an Alignment2DHorizontal and therefore without an Alignment2DVertical.

Note: it has been suggested that a more complex type of curve (e.g., BSpline) also be a supported geometry type at some future time.

### 7.5.6 Alignment for Linear Referencing

Alignment has been created as a type of PositioningElement. However, for an Alignment to properly function as such within the context of the OGC Abstract Specification Topic 19, “Linear referencing”, an additional step is necessary. If Alignment can be a linear element, it will be possible to locate things along it using linearly referenced locations.

To enable this ability, Alignment shall be specified as a type of linear element, thereby requiring Alignment to support the Topic 19 ILinearElement interface. Because Alignment has geometry/location, applications can translate between linearly referenced locations along the Alignment and spatial (x, y, and maybe z) positions. This requires that Alignment also supports the Topic 19 ISpatial interface.

<b>Requirement</b>	<p><b><a href="#">/req/alignment/LR-interfaces</a></b></p> <p>An encoding shall implement the OGC Abstract Specification Topic 19 ILinearElement and ISpatial interfaces in a manner consistent with the encoding.</p>
--------------------	--

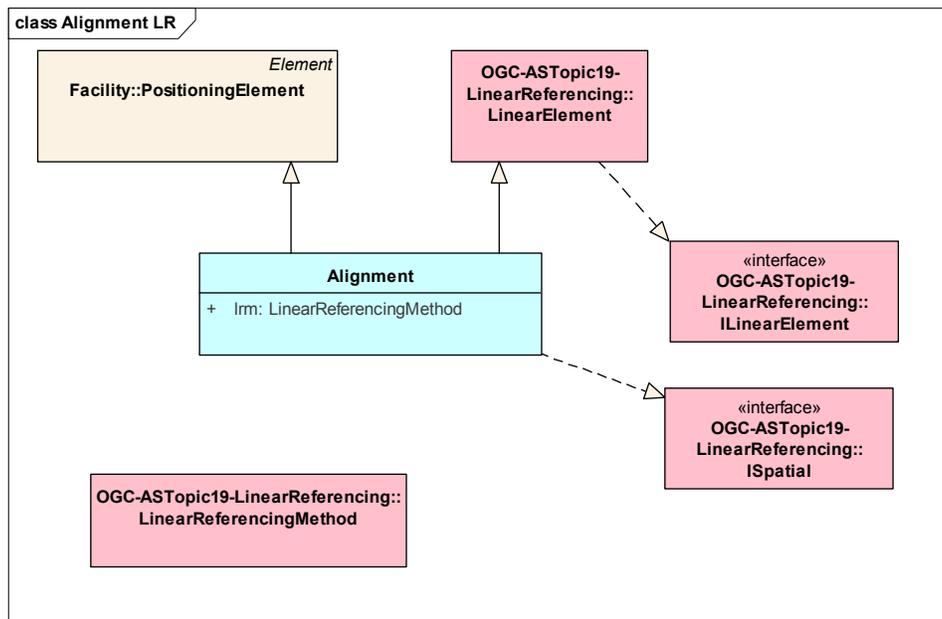


Figure 20. Alignment as a Linear Element

If an Alignment2DHorizontal.2DLineStringRepresentation or an Alignment3D.3DLineStringRepresentation is specified as a linear element, then distanceAlong shall be measured along the LineString and offsetLateralDistance and offsetVerticalDistance values shall be measured normal to the LineString from this DistanceAlong point.

If an Alignment2DHorizontal.segment list of Alignment2DHorSegments is used as a linear element, then distanceAlong and offsetLateralDistance values shall be measured in the horizontal plane, ignoring any vertical displacement of the Alignment. OffsetVerticalDistance values shall be absolute from the horizontal plane unless an Alignment2DVertical is specified as a VerticalOffsetReferent so as to take into consideration vertical displacement of the Alignment, if present.

<b>Requirement</b>	<p><b><a href="#">/req/alignment/measures</a></b></p> <p>If an Alignment2DHorizontal.2DLineStringRepresentation or an Alignment3D.3DLineStringRepresentation is specified as a linear element, then distanceAlong shall be measured along the LineString and offsetLateralDistance and offsetVerticalDistance values shall be measured normal to the LineString from this DistanceAlong point.</p> <p>If an Alignment2DHorizontal.segment list of Alignment2DHorSegments is used as a linear element, then distanceAlong and offsetLateralDistance values shall be measured in the horizontal plane, ignoring any vertical displacement of the Alignment. OffsetVerticalDistance values shall be absolute from the horizontal plane unless an Alignment2DVertical is specified as a VerticalOffsetReferent so as to take into consideration vertical displacement of the Alignment, if present.</p>
--------------------	---

Other than for defining the distance along values for Alignment2DVertical segments, linearly referenced locations specified along an Alignment are not restricted to using the (default) LRM of the Alignment.

For more information about how Alignments can be used for Linear Referencing, see Annex C: Linear Referencing below.

### 7.5.7 Alignment Summary

To summarize, an Alignment can be defined in several ways:

- 1) as a horizontal alignment: that is, a 2D projection onto a horizontal plane;
- 2) with an accompanying 2D vertical long section taken along the horizontal alignment;
- 3) with a 3D linestring generated from the horizontal and vertical alignments if they also exist; or
- 4) only as a 3D linestring.

For case 1), the Alignment only has (exactly one) Alignment2DHorizontal, specified as a 2D LineString or by an ordered set of Alignment2DHorSegments, or both.

For case 2), the Alignment only has (exactly one) Alignment2DHorizontal, specified by an ordered set of Alignment2DHorSegments, and exactly one Alignment2DVertical.

For case 3), the Alignment only has (exactly one) Alignment2DHorizontal, exactly one Alignment2DVertical, and exactly one Alignment3D. The geometry of the Alignment3D shall be consistent with the geometry of the Alignment2DHorizontal and Alignment2DVertical.

For case 4), the Alignment only has (exactly one) Alignment3D.

When an Alignment has an Alignment2DHorizontal, this Alignment2DHorizontal may be defined merely by a simple, 2dLinestringRepresentation. An Alignment2DHorizontal may also or alternatively be defined by one or more Alignment2DHorSegments.

An Alignment is a PositioningElement which provides a Linear Referencing System for locating PhysicalElements. To achieve this, Alignment must behave as a Topic 19 linear element.

## 7.6 Requirements class: Road

Requirements Class	
<b>/req/road</b>	
Target type	Encoding of the conceptual model
Name	Road
Dependency	<b>/req/land-feature</b>
Dependency	<b>/req/facility</b>
Requirement	<b>/req/road/classes</b>
Requirement	<b>/req/road/alignment</b>

The Road Requirements Class supports those use cases in which a designer wishes to exchange the output of the design with someone who is likely to use the design for purposes other than completing the road design. On the other hand, a possible future RoadDesign Requirements Class could support the more complex designer to designer information interchange, such as would exist when a designer other than the original designer takes over the design process to complete the design. Alternatively, this may be left to IFCs.

Consequently, the Road Requirements Class includes several alternative ways for representing a design such as RoadElements, 3D StringLines (aka profile views, longitudinal breaklines, long sections), and 3D surfaces and layers, as well as collections of these. The Road Requirements Class Classes are shown in Figure 21.

<b>Requirement</b>	<b>/req/road/classes</b>
	The Road Requirements Class Classes shown in blue in Figure 21 shall be provided for by the encoding in a manner consistent with the encoding.

RoadElements can be linearly located along any type of linear element, as prescribed by OGC Abstract Specification Topic 19, Linear Referencing. If the linear element is an Alignment, then the Alignment Requirements Class shall be supported.

<b>Requirement</b>	<b>/req/road/alignment</b>
	If an encoding allows the linear element used for locating RoadElements to be an Alignment, then that encoding shall support the Alignment Requirements Class.

It is anticipated that the (future proposed) RoadDesign Requirements Class will cover design information in support of those use cases which involve the exchange of design information between designers. It therefore would include DesignTemplates and SuperelevationEvents.

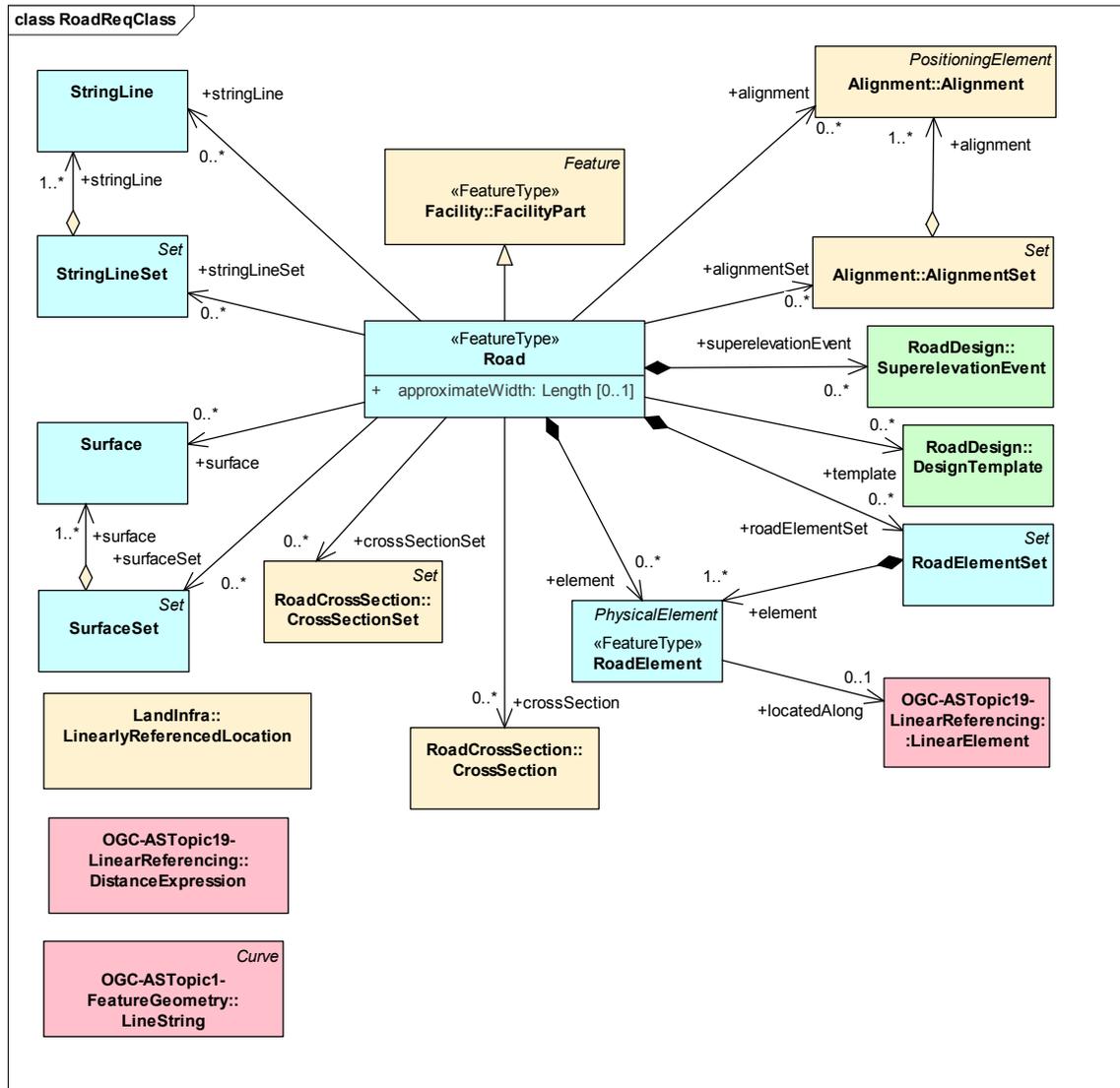


Figure 21. Road Requirements Class

### 7.6.1 Road

Road is a Feature that specifies that part of a Facility that is a single segment of road that is continuous, non-overlapping, and non-branching (though it may contain intersections with other roads).

As a subtype of Feature, Road inherits attributes featureID, name, description, spatialRepresentation, linearlyReferencedLocation, property, and propertySet. Road has the following additional attributes:

**approximateWidth:** optional, estimated width of the road used during planning or conceptual design to explore design alternatives

**alignment:** any number of Alignment PositioningElements used to define the geometry of the road

**alignmentSet:** any number of sets of Alignments grouped for a purpose

**element:** any number of PhysicalElements of type RoadElement that together make up the Road

**roadElementSet:** any number of sets of RoadElements grouped for a purpose

**stringLine:** any number of StringLine longitudinal lines each representing a particular position in the road construction, such as the right outside edge of pavement, as one travels along the Road

**stringLineSet:** any number of sets of StringLines grouped for a purpose

**surface:** any number of Surfaces, each at some level in the Road cross section (e.g., top surface, subgrade) and represented as a TIN surface

**surfaceSet:** any number of sets of Surfaces grouped for a purpose

**crossSection:** any number of CrossSection views cut across a Road at a particular location along the Road

**crossSectionSet:** any number of sets of CrossSections grouped for a purpose

When the RoadDesign Requirements Class is created, it is anticipated that Road will also have the following additional attributes:

**superelevationEvent:** any number of (tbd) SuperelevationEvents

**template:** any number of (tbd) DesignTemplates

## 7.6.2 Road Element

Road FacilityPart is a collection of zero or more RoadElements (e.g., pavement layer, base coarse, sidewalk, curb and gutter), specified separately and/or in RoadElementSets. The context diagram for RoadElement is shown in Figure 22.

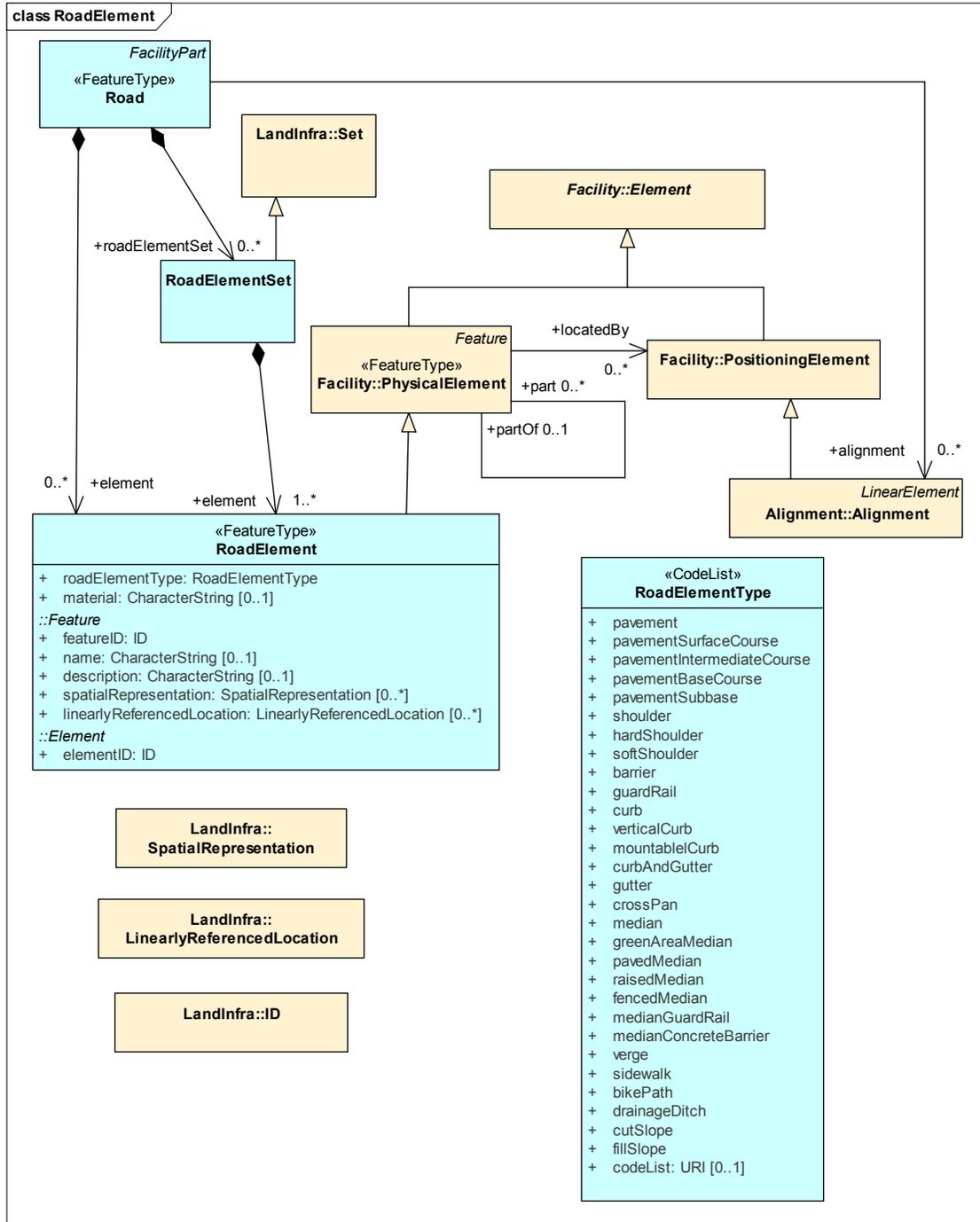


Figure 22. RoadElement context diagram

As a subtype of Feature, RoadElement inherits attributes featureID, name, description, spatialRepresentation, linearlyReferencedLocation, property, and propertySet. As a subtype of Element, RoadElement inherits attribute elementID. As a subtype of PhysicalElement, it inherits attributes part and part of. Additionally, RoadElement has the following attributes:

**roadElementType:** type of RoadElement

**material:** the optional material of the element

RoadElements can be linearly located, as specified by LinearlyReferencedLocation PositionExpressions. Part of a PositionExpression is the specification of the OGC Abstract Specification Topic 19 Linear Referencing LinearElement. This can be any type of LinearElement from Topic 19 (Feature, Curve, or DirectedEdge). An Alignment can substitute as a LinearElement since it supports the ILinear interface (see Figure 20), thereby allowing RoadElements to be linearly located along an Alignment.

The spatialRepresentation of the RoadElement is optional. If the spatialRepresentation is specified, there can be any number of these representations. For example, a 2D representation might be a Point, Curve, or Polygon Surface. In 3D, BRepSolid and PolyfaceMesh Solid representations may also be used.

### 7.6.2.1 Road Element Type

RoadElementType defines the type of RoadElement, as shown in Figure 22, e.g., pavementSurfaceCourse, curbAndGutter, or sidewalk. RoadElementType is defined as a CodeList in accordance with ISO 19103. The inclusion of the optional tagged value codeList:URI choice, is in accordance with ISO 19103. If present, then the only valid values are those from an external authority referenced code list identified by the URI value. If not, the other CodeListed values are valid along with possible additional, user defined values. However, these additional values should not replace an existing code by changing the name or definition, or have the same definition as an existing value.

### 7.6.3 Road Element Set

A RoadElementSet is a set of RoadElements grouped together for a particular reason. As a subtype of Set, it inherits the Set attributes of name, description, and authority. RoadElementSet has the following additional attribute:

**element:** one or more RoadElements

### 7.6.4 String Line

A StringLine (aka long section, longitudinal breakline) is a 3D longitudinal view of a position in the road construction, such as the right outside edge of pavement, as you travel along a Road. Any number of StringLines may be specified for a Road, either individually or in StringLineSets. The context diagram is shown in Figure 23.

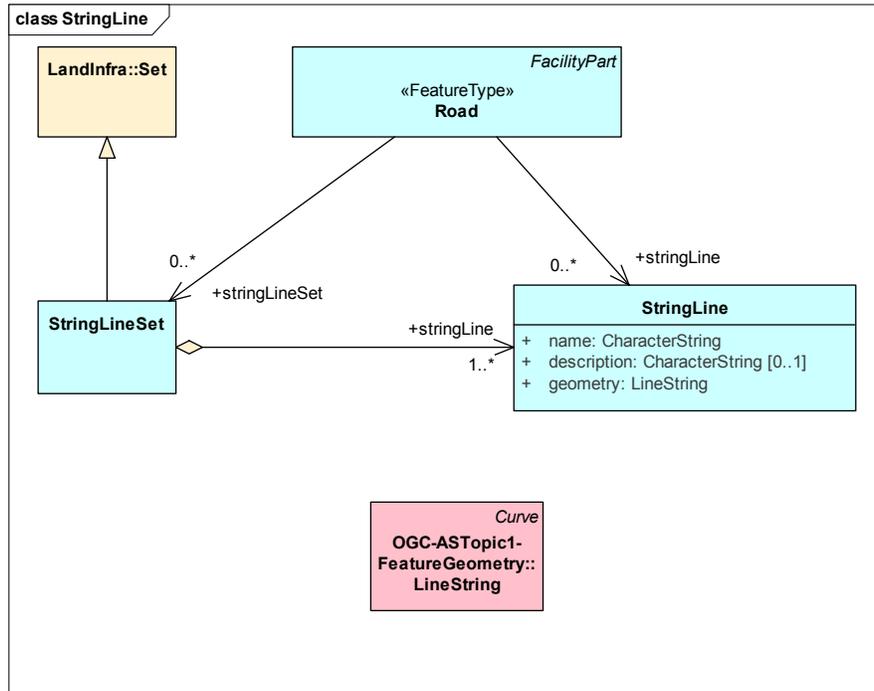


Figure 23. Road StringLine context diagram

StringLine has the following attributes:

**name:** user defined name used to identify the StringLine, unique within a Road

**description:** optional explanation of what the StringLine represents

**geometry:** the 3D LineString geometry of the StringLine

### 7.6.5 String Line Set

A StringLineSet is a set of StringLines grouped together for a particular reason. As a subtype of Set, a StringLineSet inherits the Set attributes of name, description, and authority. StringLineSet has the following additional attribute:

**stringLine:** one or more StringLines

### 7.6.6 Surface

A Surface representation of a Road is a TIN at a particular level within a Road cross section. For example, it can represent the top surface of the constructed Road or the subgrade level of the finished earthwork which supports the actual RoadElements, such as pavement and sidewalk.

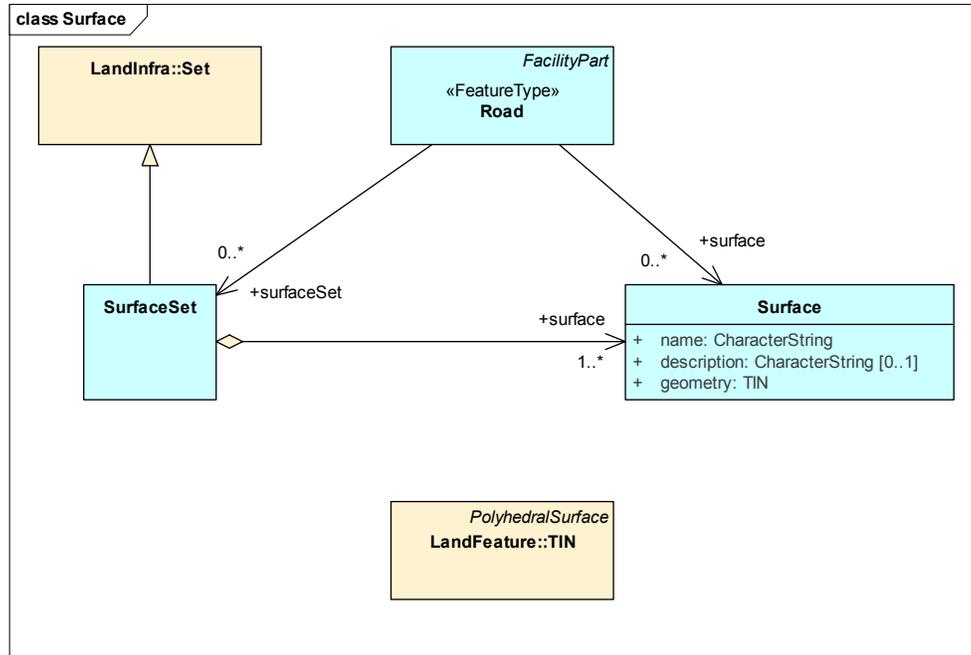


Figure 24. Road Surface context diagram

Surface has the following attributes:

**name:** user defined name used to identify the Surface, unique within a Road

**description:** optional explanation of what the Surface represents

**geometry:** the TIN geometry of the Surface

### 7.6.7 Surface Set

A SurfaceSet is a set of Surfaces grouped together for a particular reason. As a subtype of Set, SurfaceSet inherits the Set attributes of name, description, and authority.

SurfaceSet has the following additional attribute:

**surface:** one or more Surfaces

### 7.6.8 RoadCrossSection Requirements Class

Requirements Class	
<b>/req/road-cross-section</b>	
Target type	Encoding of the conceptual model
Name	Road Cross Section
Dependency	<b>/req/road</b>
Requirement	<b>/req/road-cross-section/classes</b>
Requirement	<b>/req/road-cross-section/alignment</b>

The RoadCrossSection Requirements Class extends the Road Requirements Class by adding the 2D CrossSection alternative way of representing a design, as well as collections of these.

<b>Requirement</b>	<b>/req/road-cross-section/classes</b>
	The RoadCrossSection Requirements Class Classes shown in blue in Figure 25 shall be provided for by the encoding in a manner consistent with the encoding.

CrossSections can be linearly located along any type of linear element, as prescribed by OGC Abstract Specification Topic 19, Linear Referencing. If the linear element is an Alignment, then the Alignment Requirements Class shall be supported.

<b>Requirement</b>	<b>/req/road-cross-section/alignment</b>
	If an encoding allows the linear element used for locating CrossSections to be an Alignment, then that encoding shall support the Alignment Requirements Class.

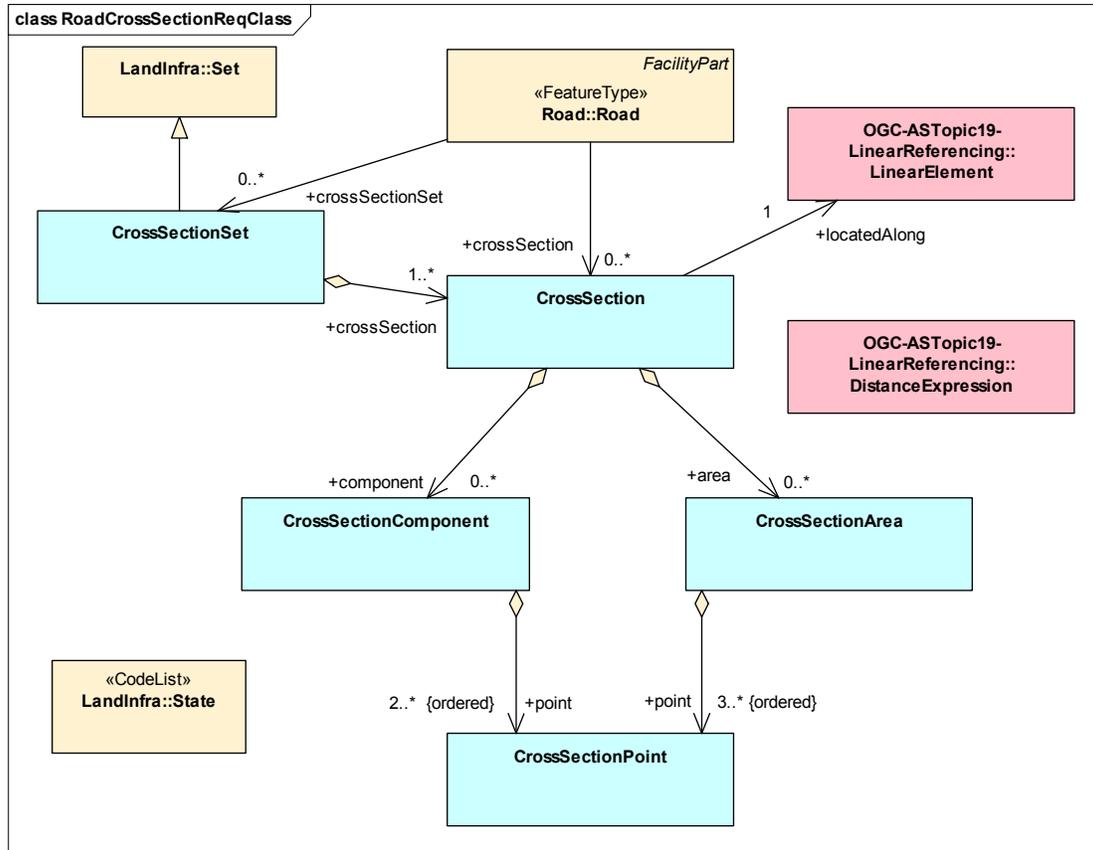


Figure 25. RoadCrossSection Requirements Class

### 7.6.8.1 Cross Section

A CrossSection describes how a Road looks (or will look), in a 2D cross section view, at some specific location along its length. CrossSections are typically an output of the Road design, rather than a set of requirements like a DesignTemplate. They may include just the resultant top surface (early design output) or may include subsurface information as well. Linear interpolation between CrossSections is assumed.

The two CrossSection axes represent the horizontal and vertical offsets for CrossSectionPoints. The axis origins represent a hook point for the CrossSection which is used to specify how the CrossSection is attached along a LinearElement. If this is an Alignment, then the horizontal offset coordinates measure distances right (positive) or left (negative) from the Alignment2DHorizontal and the vertical offset coordinates measure the distance up (positive) or down (negative) from the Alignment2DVertical at the location specified for that CrossSection. The CrossSection location is specified as an Alignment2DHorizontal distance along value. Because the existing ground surface is known at this Alignment location, the exact coordinates of the CrossSectionPoints at either end of the CrossSection can be determined and are therefore included in this view.

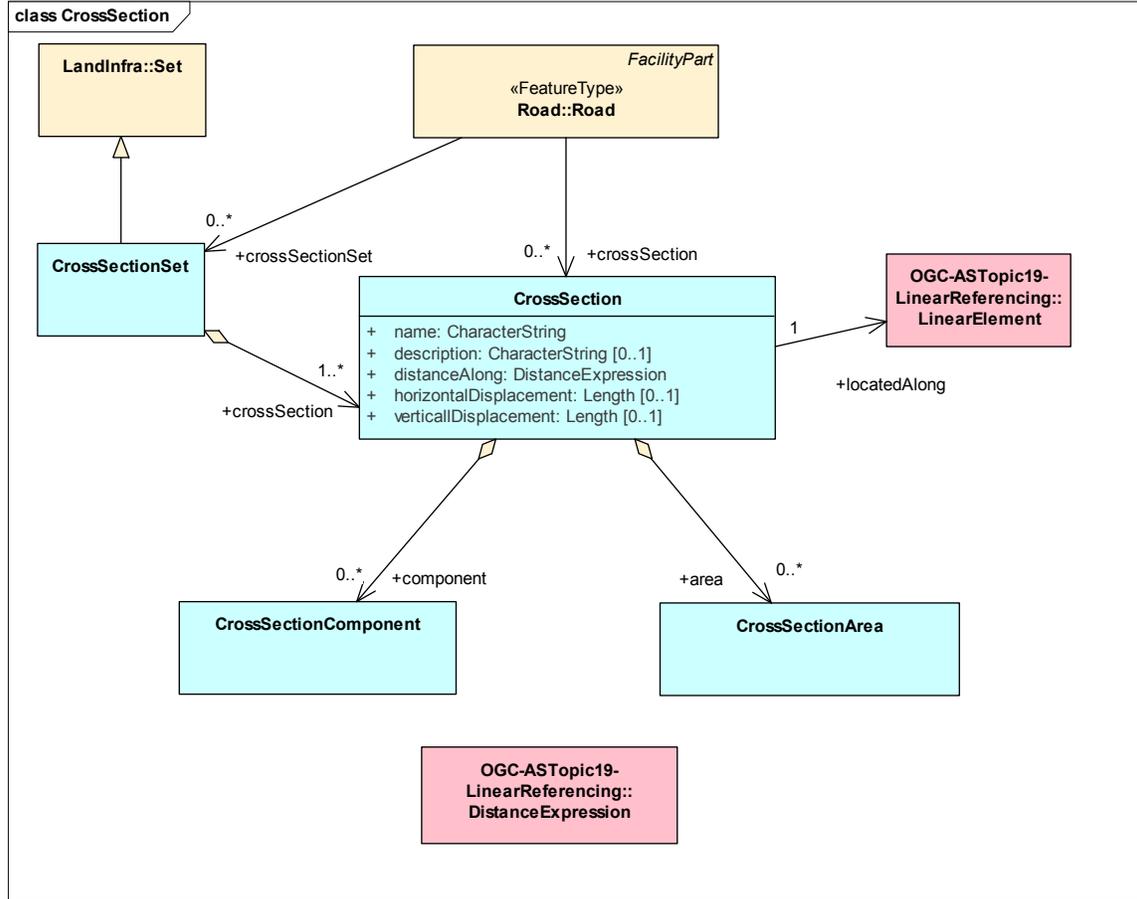


Figure 26. CrossSection context diagram

The context diagram for CrossSection is Figure 26. CrossSection has the following attributes:

**name:** user defined name used to identify the CrossSection, unique within a Road

**description:** optional explanation of the CrossSection

**distanceAlong:** the distance measured along the LinearElement to get to the CrossSection location

**horizontalDisplacement:** optional lateral distance measured left (negative) or right (positive) perpendicular from the LinearElement at the distance along location to the CrossSection origin

**verticalDisplacement:** optional vertical distance measured up (positive) or down (negative) from the Alignment to the CrossSection origin

**locatedAlong:** the LinearElement along which the CrossSection is located

**component:** any number of CrossSectionComponents

**area:** any number of CrossSectionAreas

### 7.6.8.2 Cross Section Set

A CrossSectionSet is a set of CrossSections grouped together for a particular reason. As a subtype of Set, it inherits the Set attributes of name, description, and authority.

CrossSectionSet has the following additional attribute:

**crossSection:** one or more CrossSections

### 7.6.8.3 Cross Section Component

CrossSectionComponents are logical parts of a CrossSection, represented in 2D cross section view. Portions of the CrossSection that are separated into CrossSectionComponents are typically, though not always, based on tabulation considerations. Examples of CrossSectionComponents include pavement layers, curb and gutter, sidewalk, ditches, and cut and fill slopes. Consequently, these CrossSectionComponents typically represent the RoadElements for the Road.

Figure 27 shows a (non-normative example) CrossSection for a two lane Road with pavement over base course layers and having a gravel and unpaved shoulder, ditch and cut slope on the left side and curb and gutter, sidewalk, verge, and fill slope on the right side. The CrossSectionComponents which comprise the CrossSection are so labelled.

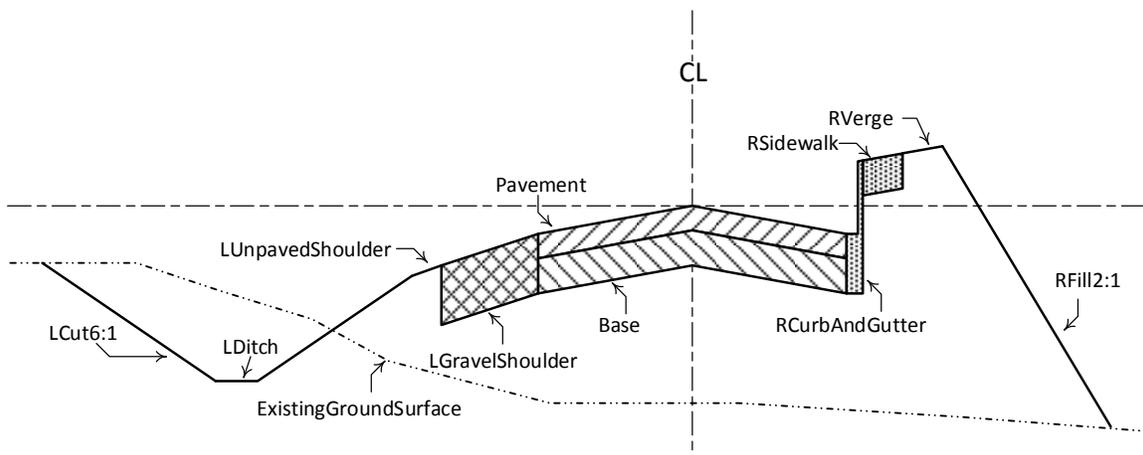


Figure 27. Two Lane CrossSection

Figure 28 shows the complete CrossSection with horizontal dimensions (in meters) and appropriate cross slopes.

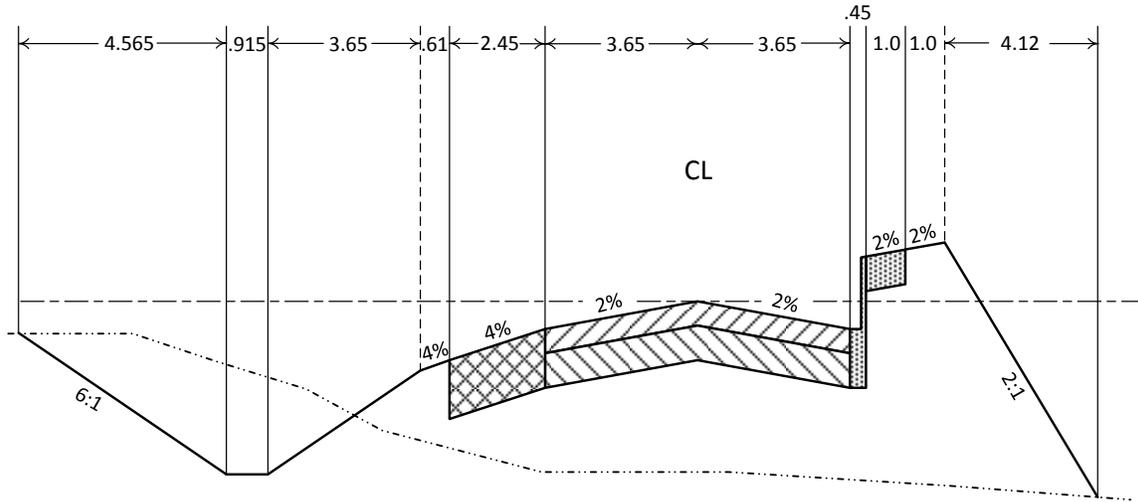


Figure 28. CrossSection horizontal dimensioning (meters) and cross slopes

The shape of CrossSectionComponents is specified by 2D CrossSectionPoints. The CrossSectionComponents which represent the pavement, base, and left gravel shoulder are shown in Figure 29 with their CrossSectionPoint names and vertical dimensions. The names are abbreviations for their description, e.g., REP is for “right edge of pavement”. The full list of CrossSectionPoints for this CrossSection are shown in Table 2 in 7.6.8.4 with their short names and full descriptions. The dimensions are used to calculate the CrossSectionPoint horizontal and vertical offsets also shown in Table 2.

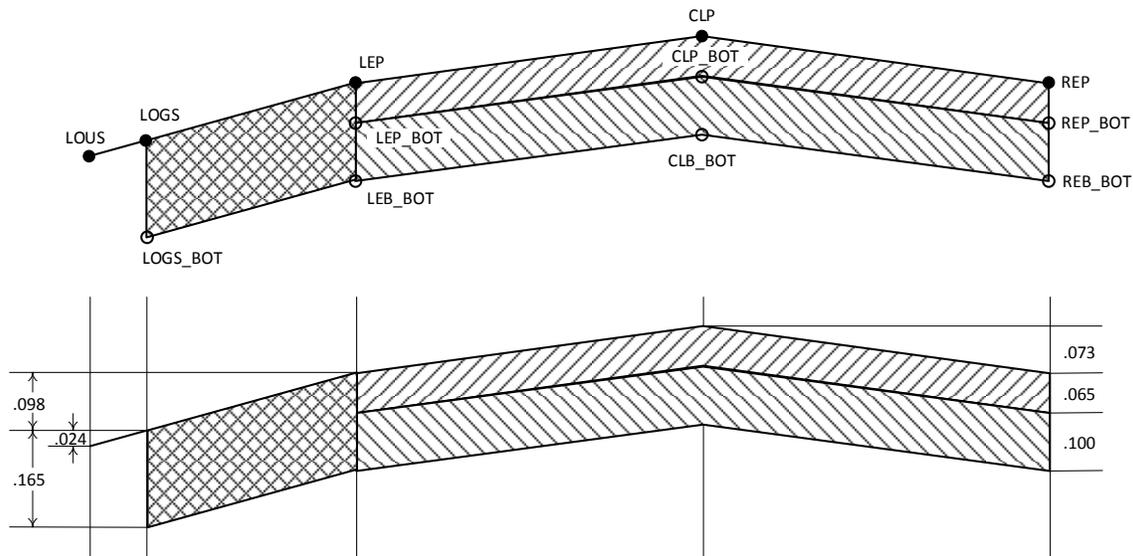


Figure 29. Pavement, base, and left gravel shoulder CrossSectionPoints and vertical dimensions (meters)

The CrossSectionComponent which represents the curb and gutter has supplemental dimensions and CrossSectionPoints as shown in Figure 30.

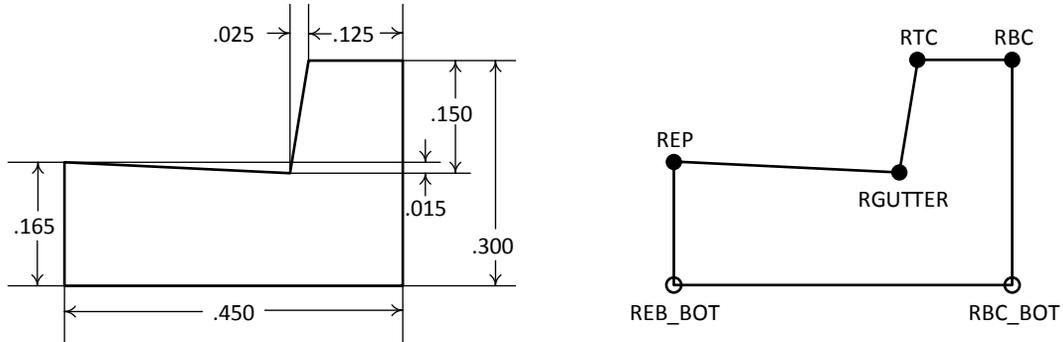


Figure 30. Curb and gutter supplemental dimensions and CrossSectionPoints

The CrossSectionComponents which represent the sidewalk, verge, and fill slope on the right hand side have CrossSectionPoints and supplemental dimensions as shown in Figure 31.

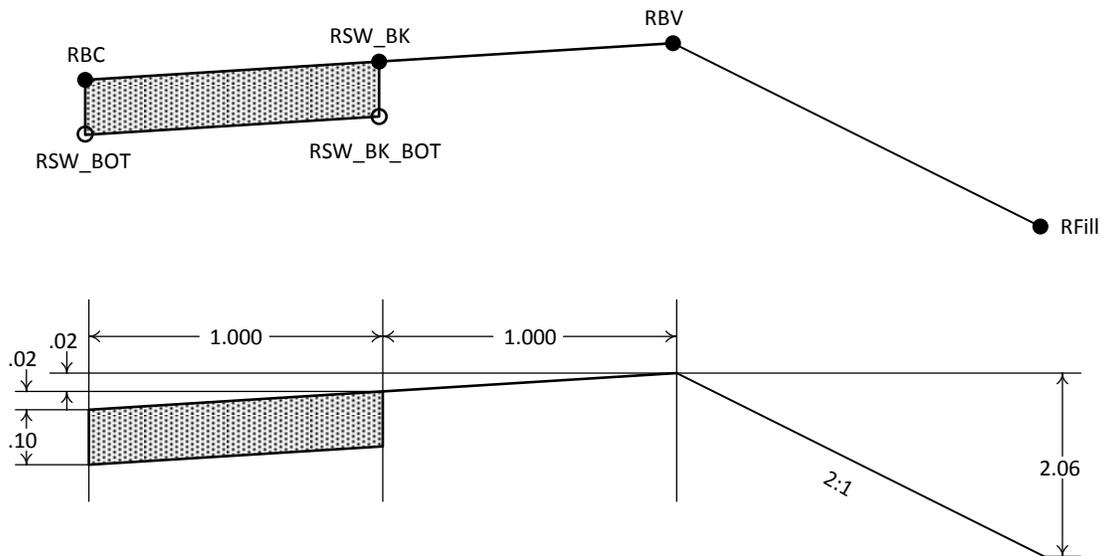


Figure 31. Right side sidewalk, verge, and fill slope CrossSectionPoints and supplemental dimensions (meters)

The CrossSectionComponents which represent the ditch and cut slope on the left hand side have CrossSectionPoints and supplemental dimensions as shown in Figure 32.

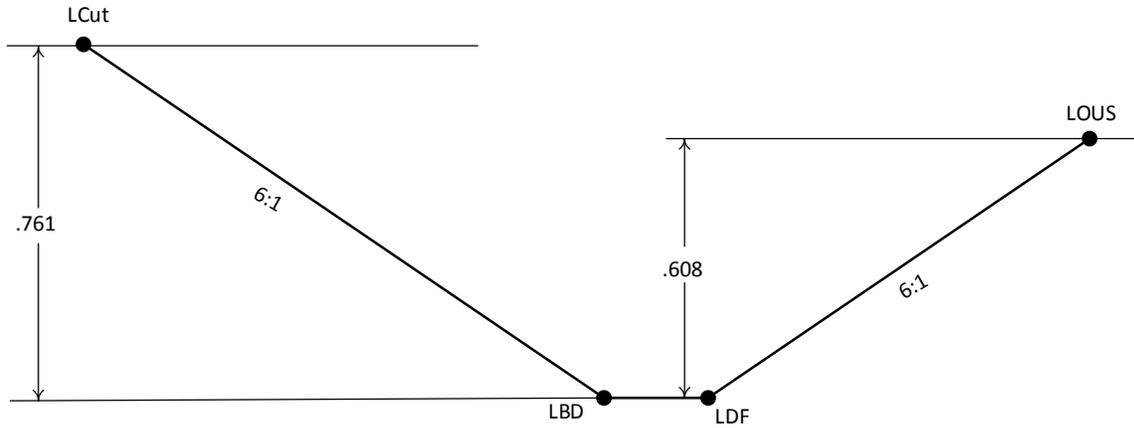


Figure 32. Left side ditch and cut slope CrossSectionPoints and supplemental dimensions (meters)

The context diagram for CrossSectionComponent and CrossSectionPoint is shown in Figure 33.

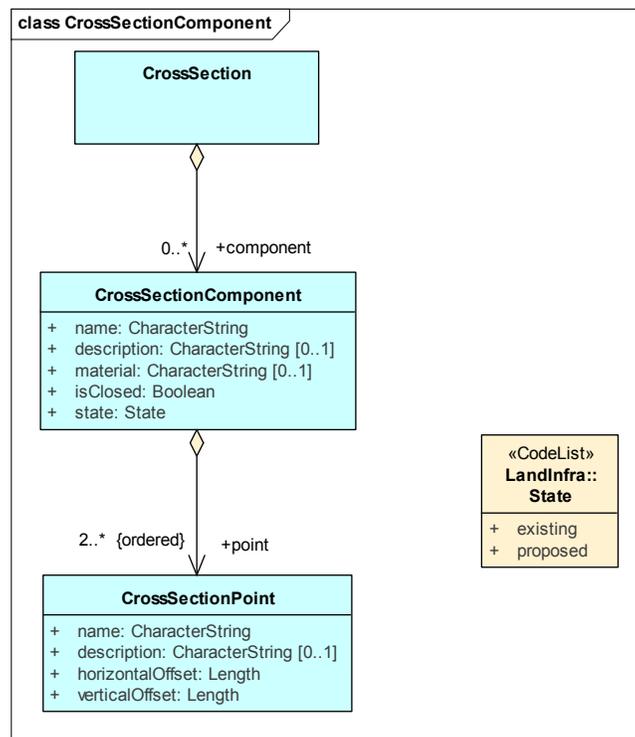


Figure 33. CrossSectionComponent and CrossSectionPoint context diagram

CrossSectionComponent has the following attributes:

- name:** user defined name used to identify the component, unique within a CrossSection
- description:** optional explanation for the component
- material:** the optional material used to construct the component
- isClosed:** True if the CrossSectionPoints form a closed area (e.g., pavement, gravel shoulder, curb and gutter, sidewalk); False if the CrossSectionPoints represent just a (top) surface (e.g., unpaved shoulder, verge, ditch, cut/fill slopes)
- state:** whether the component is existing or proposed
- point:** two or more CrossSectionPoints which define the geometry of the component in “connect the points” order

The state flag allows for CrossSectionComponents which represent existing as well as proposed components. The existing top ground surface or any material layer can therefore be specified as an existing CrossSectionComponent.

For CrossSectionComponents with isClosed = False, it is recommended that the order of the CrossSectionPoints begin with the innermost point if the component is right or left of the origin but with the left-most point of the component straddles the origin. If isClosed = True, it is recommended that the order of the CrossSectionPoints be clockwise starting with the innermost, topmost point if straddling or right of the origin and counterclockwise starting with the innermost, topmost point if left of the origin. The starting point should not be repeated as the last point, as closure will be automatic for closed components.

The proposed CrossSectionComponents which comprise the example two lane CrossSection are shown in Table 1.

name	points	isClosed
Pavement	CLP REP REP_BOT CLP_BOT LEP_BOT LEP	T
Base	CLP_BOT REP_BOT REB_BOT	T

	CLB_BOT LEB_BOT LEP_BOT	
RCurbAndGutter	REP RGUTTER RTC RBC RBC_BOT REB_BOT	T
RSidewalk	RBC RSW_BK RSW_BK_BOT RSW_BOT	T
RVerge	RSW_BK RBV	F
RFill2:1	RBV RFill	F
LGravelShoulder	LEP LOGS LOGS_BOT LEB_BOT	T
LUnpavedShoulder	LOGS LOUS	F
LDitch	LOUS LDF LBD	F
LCut6:1	LBD LCut	F

*Table 1. CrossSectionComponents*

#### 7.6.8.4 Cross Section Point

CrossSectionPoints are 2D points contained within a CrossSection view which help define the geometry of CrossSectionComponents. A local Cartesian coordinate system is established for the CrossSection with origin (0,0) being the location of the CrossSection hook point. CrossSectionPoints have coordinate values representing the known horizontal and vertical offset displacements from this origin.

The context diagram for CrossSectionPoint is shown in Figure 33. CrossSectionPoint has the following attributes:

**name:** user defined name used to identify the point, unique within a CrossSection

**description:** optional explanation of what the point represents

**horizontalOffset:** distance measured horizontally right (positive) or left (negative) from the CrossSection hook point

**verticalOffset:** distance measured vertically up (positive) or down (negative) from the CrossSection hook point

Care should be taken in selecting CrossSectionPoint names. CrossSection Point names shall be unique within a single CrossSection. However, use of consistent naming across all of the CrossSections used for a Road, e.g., “rightEdgeOfPavement” is advisable. These similarly named CrossSectionPoints can be connected up along the Road, resulting in a longitudinal breaklines (long section or profile) having the same name as the contributing CrossSectionPoints. The “rightEdgeOfPavement” CrossSectionPoints in consecutive CrossSections along the Road are thus connected together to create a rightEdgeOfPavement longitudinal breakline feature in 3D.

The CrossSectionPoints used for the CrossSectionComponents in section 7.6.8.2 are shown in Table 2.

name	description	horizontal	vertical
CLP	centerline pavement	0.000	0.000
REP	right edge of pavement	3.650	-0.073
REP_BOT	right edge of pavement bottom	3.650	-0.138
CLP_BOT	centerline pavement bottom	0.000	-0.065
LEP	left edge of pavement	-3.650	-0.073
LEP_BOT	left edge of pavement bottom	-3.650	-0.138
REB_BOT	right edge of base bottom	3.650	-0.238
CLB_BOT	centerline base bottom	0.000	-0.165
LEB_BOT	left edge of base bottom	-3.650	-0.238
LOGS	left outside gravel shoulder	-6.100	-0.171
LOGS_BOT	left outside gravel shoulder bottom	-6.100	-0.336
LOUS	left outside unpaved shoulder	-6.710	-0.195
RGUTTER	right gutter	3.950	-0.088
RTC	right top of curb	3.975	0.062
RBC	right back of curb	4.100	0.062
RBC_BOT	right back of curb bottom	4.100	-0.238
RSW_BK	right sidewalk back	5.100	0.082
RSW_BK_BOT	right sidewalk back bottom	5.100	-0.018
RSW_BOT	right sidewalk bottom	4.100	-0.038
RBV	right back of verge	6.100	0.102

RFill	2:1 fill	8.100	-0.898
LDF	left ditch foreslope	-10.360	-0.803
LBD	left bottom ditch	-11.275	-0.803
LCut	6:1 cut	-15.840	-0.042

Table 2. CrossSectionComponent CrossSectionPoints

### 7.6.8.5 Cross Section Area

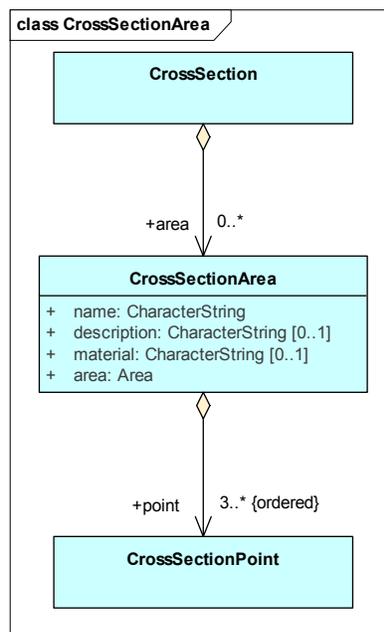


Figure 34. CrossSectionArea context diagram

CrossSectionAreas are logical parts of a CrossSection, represented in 2D cross section view. Portions of the CrossSection that are separated into CrossSectionAreas are normally based on tabulation considerations. Examples of CrossSectionAreas include areas of cut and fill of a particular kind of material. CrossSectionAreas differ from CrossSectionComponents in that the former are typically bounded by multiple CrossSectionComponents. For example, the CrossSectionArea FILL in Figure 34 is bounded on top by the CrossSectionComponents LDitch, LUnpavedShoulder, LGravelShoulder, Base, etc.

CrossSectionArea has the following attributes:

**name:** user defined name used to identify the CrossSectionArea, unique within a CrossSection

**description:** optional explanation of the CrossSectionArea

**material:** the CrossSectionArea's optional material which can be general such as cut or fill or more specific such as a type of soil or rock

**area:** the cross sectional area of the CrossSectionArea

**point:** three or more CrossSectionPoints which define the geometry of the CrossSectionArea in "connect the points" order

Figure 35 shows the (non-normative example) CrossSection for the two lane Road from Figure 27 with cut and fill CrossSectionAreas instead of CrossSectionComponents. It also shows the additional CrossSectionPoints that would be required to specify the two CrossSectionAreas.

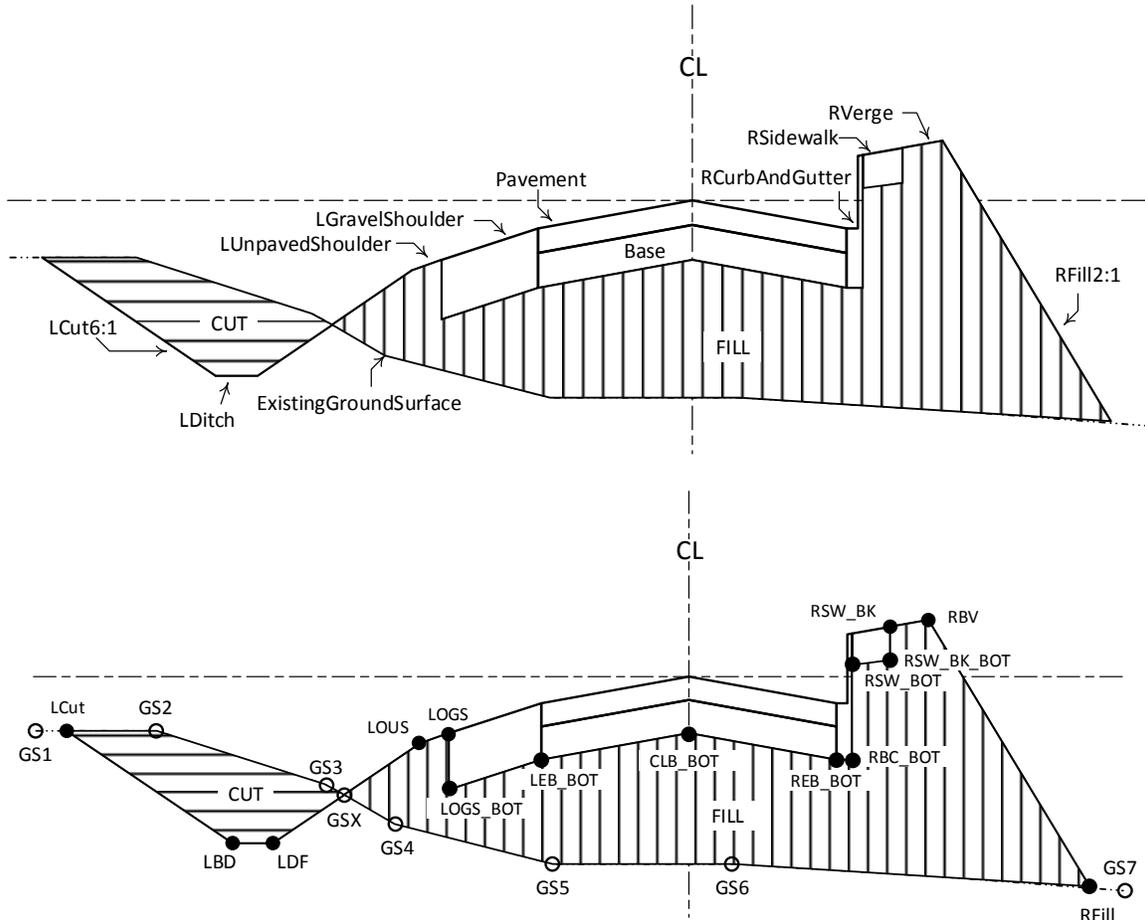


Figure 35. Two Lane Road with CUT and FILL CrossSectionAreas with additional CrossSectionPoints

## 7.7 Requirements Class: Railway

Requirements Class	
<b>/req/railway</b>	
Target type	Encoding of the conceptual model
Name	Railway
Dependency	<b>/req/facility</b>
Requirement	<b>/req/railway/classes</b>
Requirement	<b>/req/railway/alignment</b>
Requirement	<b>/req/railway/cant</b>

The Railway Requirements Class supports those use cases where a designer wishes to exchange the output of the design with someone who is likely to use it for purposes other than further design. On the other hand, a possible future RailwayDesign Requirements Class could support the more complex designer to designer information interchange, such as would exist when a designer other than the original designer takes over the design process to complete the design. Alternatively, this may be left to IFCs.

Consequently, the Railway Requirements Class covers design output such as RailwayElements, track geometry including chainage discontinuities and superelevation (cant). The Railway Requirements Class Classes are shown in Figure 36.

<b>Requirement</b>	<b>/req/railway/classes</b>
	The Railway Requirements Class Classes shown in blue in Figure 36 shall be provided for by the encoding in a manner consistent with the encoding.

RailwayElements and CantEvents can be linearly located along any type of linear element, as prescribed by OGC Abstract Specification Topic 19, Linear Referencing. If the linear element is an Alignment, then the Alignment Requirements Class shall be supported.

<b>Requirement</b>	<b>/req/railway/alignment</b>
	If an encoding allows the linear element used for locating RailwayElements or CantEvents to be an Alignment, then that encoding shall support the Alignment Requirements Class.

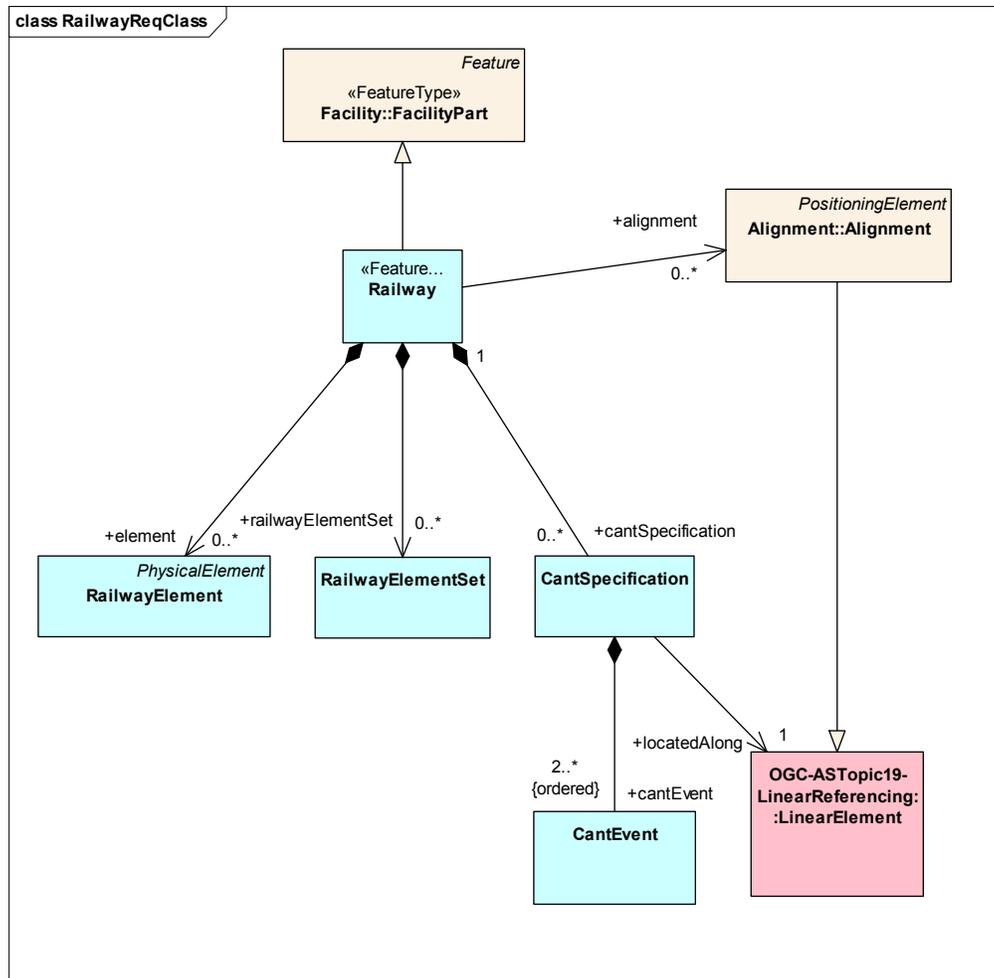


Figure 36 - Railway Requirements class

### 7.7.1 Railway

Railway is a Feature that specifies that part of a Facility that is a single segment of railway which is continuous, non-overlapping, and non-branching (though it may contain intersections with other railways). In addition to the attributes it inherits from Feature, Railway has the following attributes:

**alignment:** any number of Alignment PositioningElements used to define the geometry of the railway

**alignmentSet:** any number of sets of Alignments grouped for a purpose

**element:** any number of PhysicalElements of type RailwayElement that together make up the Railway

**railwayElementSet:** any number of sets of RailwayElements grouped for a purpose

**cantSpecification:** any number of cantSpecifications

### **7.7.2 Alignment extensions**

The alignment model described in clause 7.5 Requirements class: Alignment covers the alignment requirements for Railway.

#### **7.7.2.1 Transition curves**

The alignment class supports transition curves of the proposed ISO 19107 Spiral type. Clothoid curves are defined as a special case of curve since this is commonly used.

#### **7.7.2.2 Gradient points**

When defining vertical alignments, LandXML allows a way to specify these using PVI (Point of Vertical Intersection). A joint decision with the buildingSMART International IFC Alignment project team was to only allow for segment-, or element-, based definition of both horizontal and vertical alignments. The primary reasons for this are:

- Don't allow multiple ways to define the same thing;
- Element based definitions are more accurate and less error prone; and
- When needed, it is relatively easy to derive PVI's from elements by intersecting the straight line elements.

### 7.7.3 RailwayElement

Railway FacilityPart is a collection of zero or more RailwayElements. The context diagram for RailwayElement is shown in Figure 37.

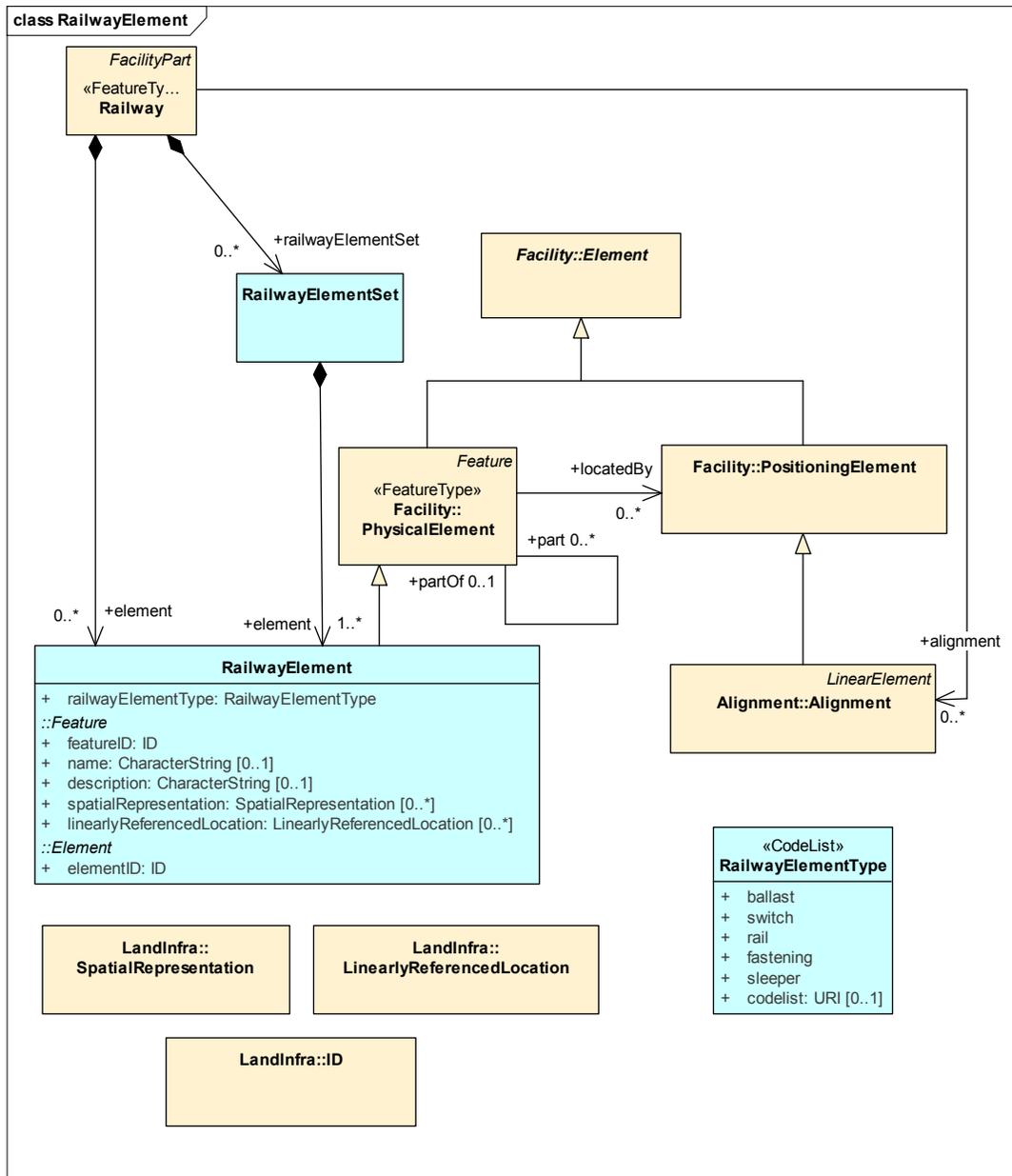


Figure 37. RailwayElement context diagram

As a subtype of Feature, RailwayElement inherits attributes featureID, name, description, spatialRepresentation, linearlyReferencedLocation, property, and propertySet. As a subtype of Element, RailwayElement inherits attribute elementID. As a subtype of PhysicalElement, the RailwayElement inherits the capability of describing decomposition hierarchical structures using the part- and partOf association roles.

Additionally, RailwayElement has the following attribute:

**railwayElementType:** the type of RailwayElement

RailwayElements can be linearly located, as specified by LinearlyReferencedLocation PositionExpressions. Part of a PositionExpression is the specification of the OGC Abstract Specification Topic 19 Linear Referencing LinearElement. This can be any type of LinearElement from Topic 19 (Feature, Curve, or DirectedEdge). An Alignment can substitute as a LinearElement since it supports the ILinear interface (see Figure 20), thereby allowing RailwayElements to be linearly located along an Alignment.

The spatialRepresentation of the RailwayElement is optional. If it is specified, there can be any number of these. For example, a 2D representation might be a Point, Curve, or Polygon Surface. In 3D, BRepSolid and PolyfaceMesh Solid representations may also be used.

### 7.7.3.1 Railway Element Type

RailwayElementType defines the type of RailwayElement, as shown in Figure 37, e.g., ballast, switch, sleeper, or rail. RailwayElementType is defined as a CodeList in accordance with ISO 19103. The inclusion of the optional tagged value codeList:URI choice, is in accordance with ISO 19103. If present, then the only valid values are those from an external authority referenced code list identified by the URI value. If not, the other CodeListed values are valid along with possible additional, user defined values. However, these additional values should not replace an existing code by changing the name or definition, or have the same definition as an existing value.

### 7.7.4 Railway Element Set

A RailwayElementSet is a set of RailwayElements grouped together for a particular reason. As a subtype of Set, it inherits the Set attributes of name, description, and authority. RailwayElementSet has the following additional attribute:

**element:** one or more RailwayElements

### 7.7.5 Cant (Railway superelevation)

The cant of a railway track (also referred to as superelevation) is the difference in elevation (height) between the two rails. This is normally done where the railway is curved; raising the outer rail providing a banked turn. The superelevation development length is the distance over which superelevation is developed from the normal cross-fall to full superelevation. This can be achieved within the transition curve which precedes the circular curve. Transition curves are used between tangent track and circular curves or between two adjacent curves to allow a gradual change in curvature and lateral acceleration. The center line of a transition curve has the same tangent at the connecting points as the adjacent part, whereas the curvature changes gradually from the value of one connection point to the value of the other.

In principle, assuming that appliedCant is 0 when the railway track is a straight line, it could be possible to only describe CantEvents when there is a cant break (change in cant at horizontal tangential points). However, to enforce clarity and allow for error checking, a collection of CantEvents shall cover the entire linear element from start to end. This collection of CantEvents is called a CantSpecification. A CantSpecification must have at least two CantEvents along its length, one at the start and one at the end. In addition to this, a CantEvent is always required when there is a cant break.

<b>Requirement</b>	<b><a href="#">/req/railway/cant</a></b>
	An encoding shall specify as a requirement that each CantSpecification shall cover the entire linear element the CantSpecification is located along from the linear element's start to its end, and that the CantSpecification shall include CantEvents at the start and end of the linear element and wherever else there is a cant break along the linear element.

The cant model uses the same principles as the Alignment model, i.e. capturing the resulting design and not the design parameters such as design speed. In addition, consistent with the alignment model, only one way of specifying cant has been selected, i.e. the difference in elevation between the rails. Other ways of specifying the same concept, e.g. slope, have to be derived.

LandXML uses Cant and CantStation instead of CantSpecification and CantEvent, respectively. LandXML states that an Alignment “owns” the Cant. In this LandInfra Standard, a CantSpecification is owned by a Railway. LandInfra requires that all CantEvents belonging to a CantSpecification be located along one and only one linear element, which in most cases will be a Railway Alignment. Since a Railway can be associated with many linear elements, a CantSpecification may be specified for each linear element, hence the zero to many multiplicity for the cantSpecification association role for Railway.

To allow for a generic approach for cant definitions that is valid for multiple life cycle stages, the cant events may be associated with any kind of linear element, e.g. an Alignment or a linear track segment in a railway network database perhaps using kilometer post referents.

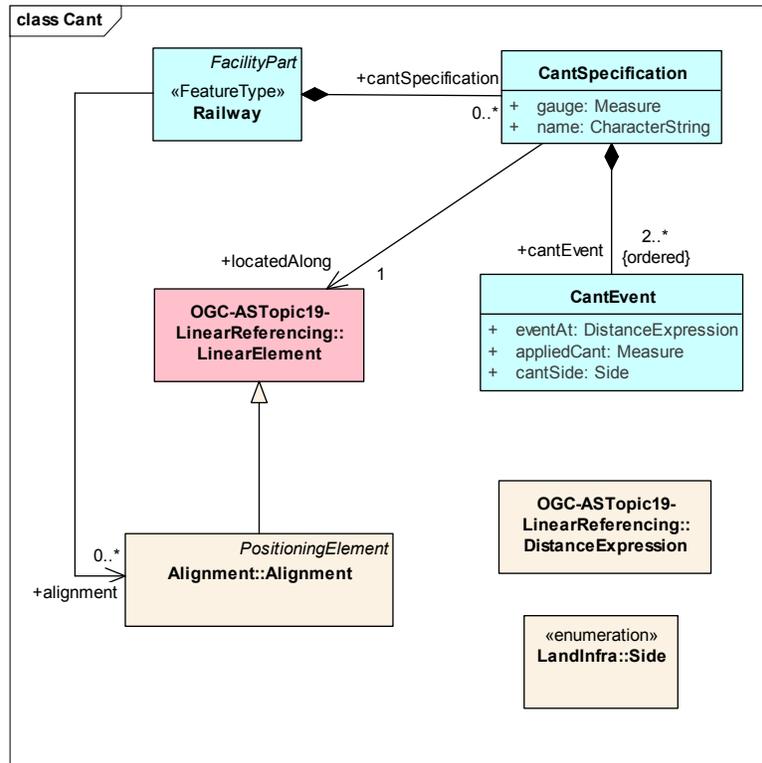


Figure 38 - Cant context diagram

### 7.7.5.1 Cant Specification

CantSpecification is a collection of CantEvents along a single linear element for a single Railway. CantSpecification has the following attributes:

- name:** name of the CantSpecification
- gauge:** measure defining the spacing between the inner faces of the load bearing rails.
- locatedAlong:** the linear element along which the CantSpecification is made
- cantEvent:** two or more cantEvents which shall be ordered in the direction of the linear element

### 7.7.5.2 Cant Event

CantEvent marks a location along the linear element where an applied cant value is specified. If two succeeding CantEvents have the same applied cant value, then that value applies to the entire linear element segment between those two events. If two succeeding CantEvents have different applied cant values, then the cant value is assumed to vary linearly between the two events. CantEvent has the following attributes:

**eventAt:** DistanceExpression defining the linearly referenced location for the CantEvent as a single point along the linear element

**appliedCant:** difference in elevation between the two rails at the eventAt location

**cantSide:** left or right rail (or both) to which the cant shall be applied. The side relates to the positive direction of the linear element

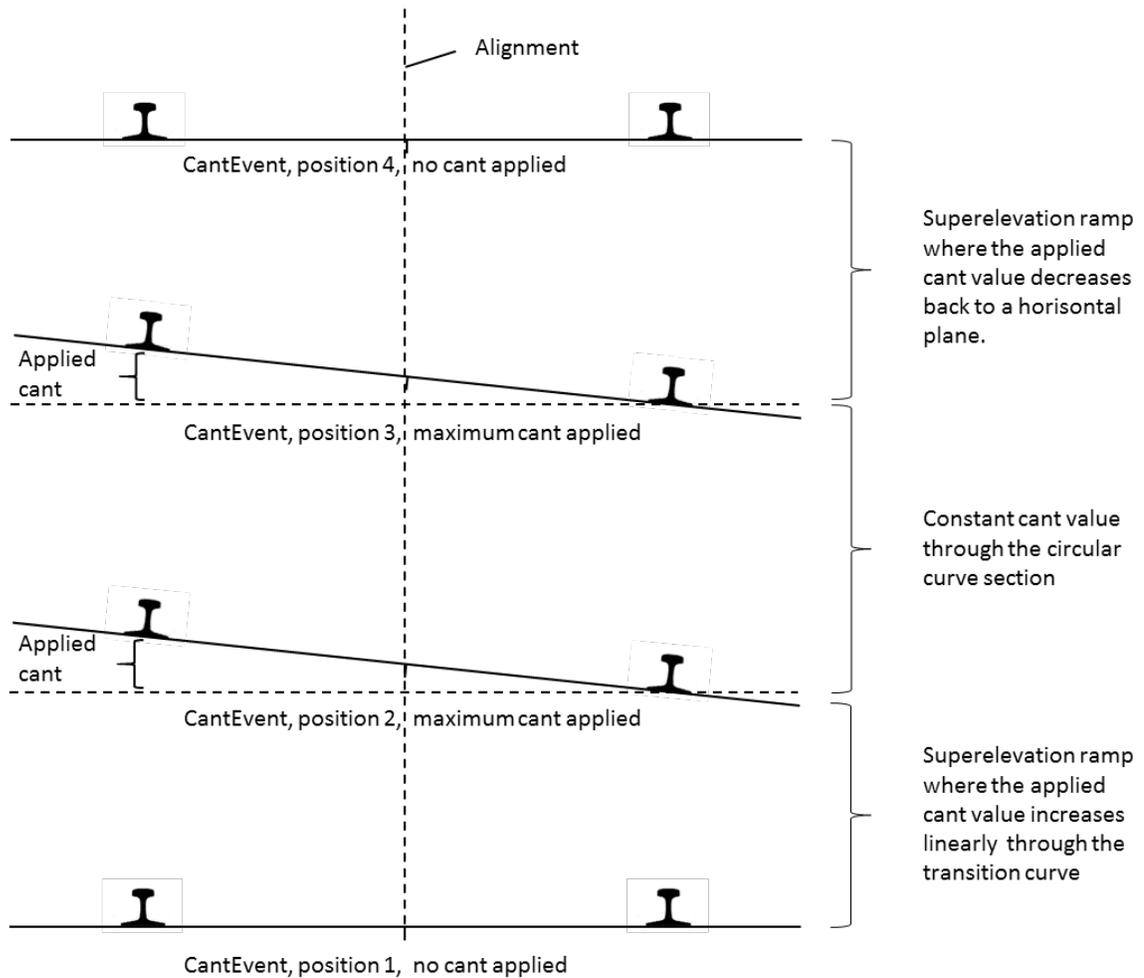


Figure 39 - Cross section of left-sided CantEvents

In this example, position 1 marks the first CantEvent. The rails are at the same elevation so the applied cant value is zero.

Between position 1 and position 2, the left rail is linearly elevated through the horizontal alignment transition curve. The right rail stays at its original zero position.

The CantEvent at position 2 marks the point along the linear element where the maximum applied cant has been achieved. This is where the superelevation ramp between position 1 and position 2 finishes. The horizontal alignment transition curve changes to a circular curve with at constant cant value until position 3.

The CantEvent at position 3 marks the point along the rail where the circular curve passes to a transition curve.

Between position 3 and position 4, the cant for the left rail is linearly decreased through the transition curve. The right rail stays at its original zero position.

The CantEvent at position 4 marks the end of the transition curve. The cant value is back to zero as the two rails once again are at the same elevation.

## 7.8 Requirements class: Survey

Requirements Class	
<b>/req/survey</b>	
Target type	Encoding of the conceptual model
Name	Survey
Dependency	<b>/req/land-infra</b>
<b>Requirement</b>	<b>/req/survey/classes</b>

A LandInfra dataset may contain any number of Surveys to cover the acquisition of points, lines, surfaces and properties of features of interest. The primary focus of this package is to have the possibility of recording and reprocessing the observations of the acquired objects.

The present Survey class contains header information for the surveys, as the survey package has been divided in sub- packages because of the number of classes in the Observations, SurveyResults and Equipment packages.

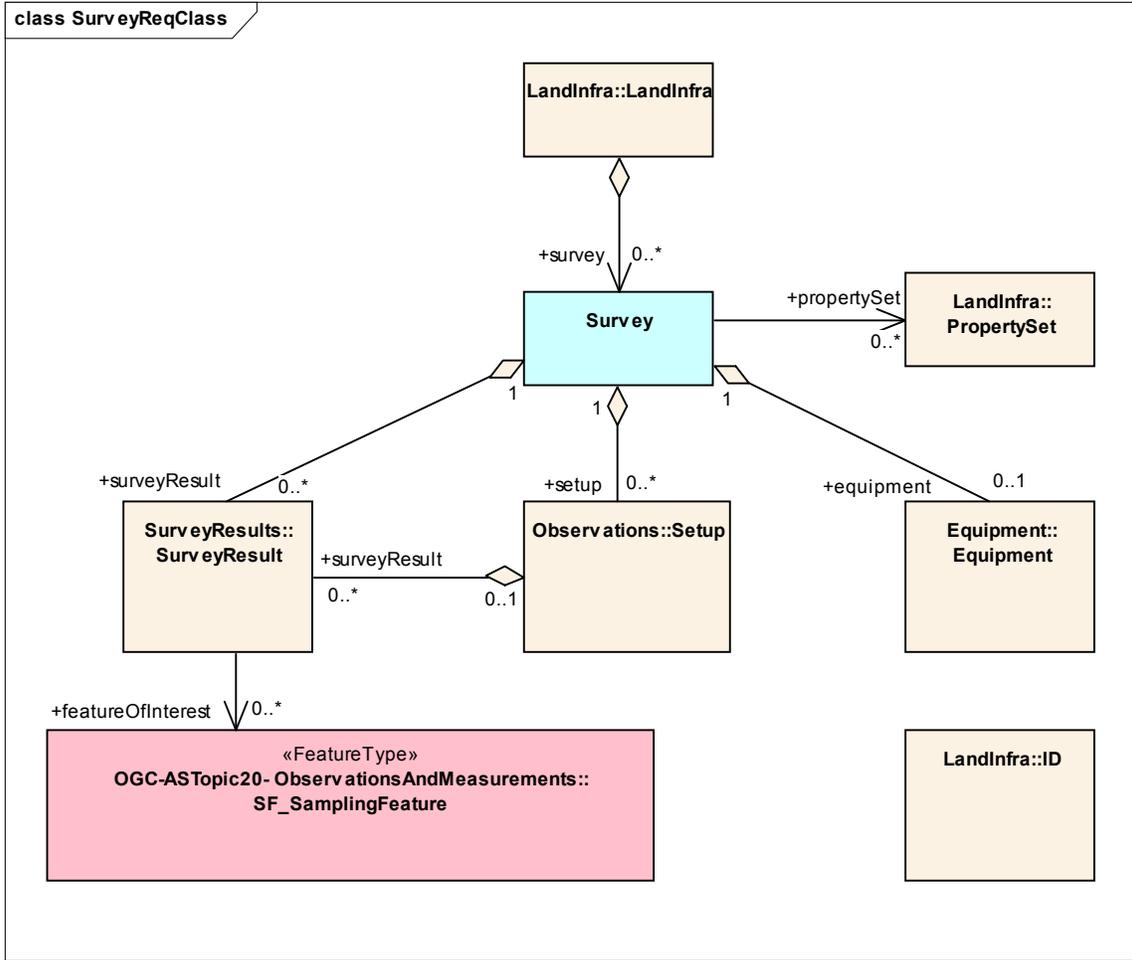


Figure 40. Survey Requirements Class

### 7.8.1 Survey Requirements Class Classes

<b>Requirement</b>	<b><a href="/req/survey/classes">/req/survey/classes</a></b>
	The Survey Requirements Class Classes shown in blue in Figure 41 shall be provided for by the encoding in a manner consistent with the encoding.

### 7.8.1.1 Survey

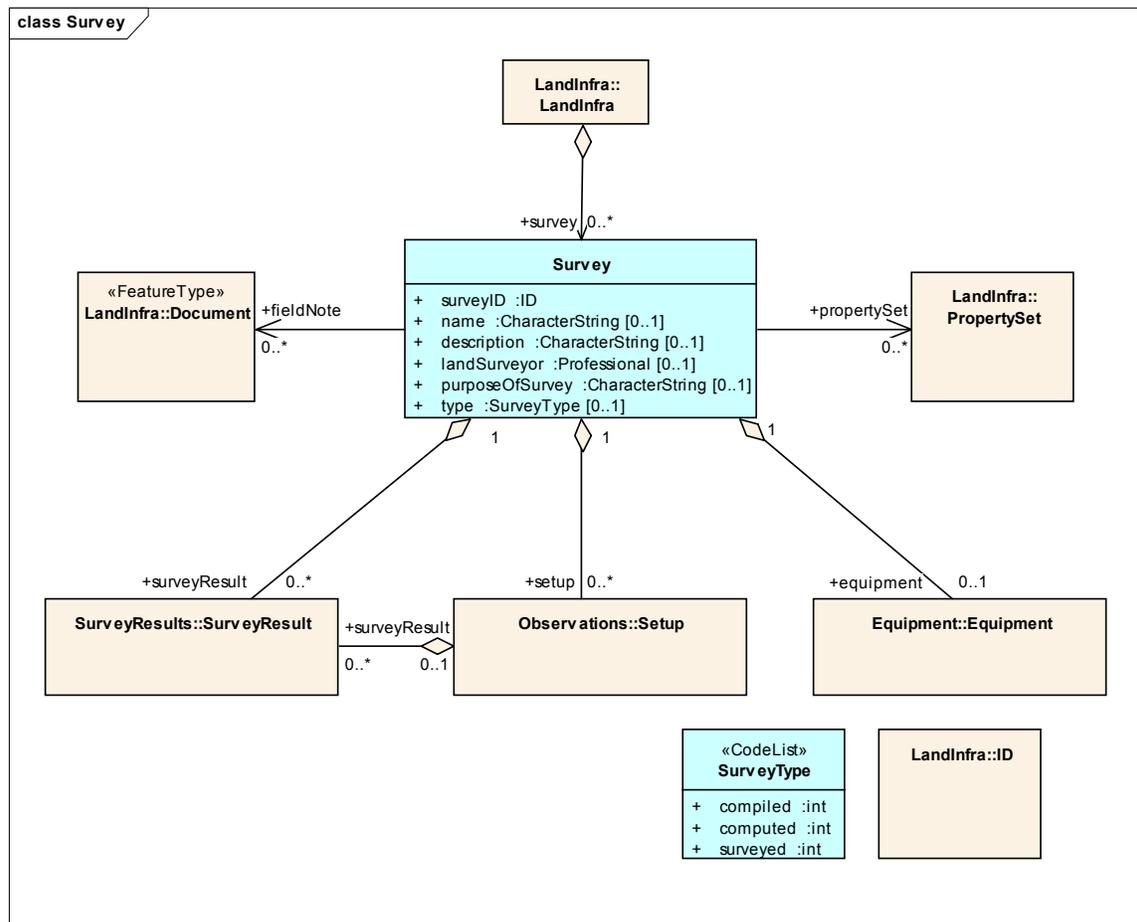


Figure 41. Survey Class

A LandInfra dataset may contain any number of Surveys. Surveys have the following attributes:

**surveyID:** user defined unique ID used to identify a survey

**name:** optional

**description:** optional

**landSurveyor:** information about the surveyor (name, company name, registration, reference)

**purposeOfSurvey:** This attribute details the purpose of survey (i.e. Land parcel subdivision, cf. Annex B.5.3) and is an enumerated list which is jurisdiction specific. A survey may have more than one purpose, as specified by the jurisdiction-specific business rules.

**type:** This element identifies if the survey has been surveyed, compiled or computed.

### **7.8.1.2 SurveyType**

This enumeration indicates whether the survey was actually performed in the field, compiled from a series of existing surveys, or simply computed using known observations and math.

## 7.8.2 Requirements class: Observations

Requirements Class	
<b>/req/observations</b>	
Target type	Encoding of the conceptual model
Name	Observations
Dependency	<b>/req/survey</b>
Dependency	<a href="http://www.opengis.net/doc/is/om/2.0">http://www.opengis.net/doc/is/om/2.0</a>
Dependency	<a href="urn:iso:is:iso:19115:clause:7">urn:iso:is:iso:19115:clause:7</a>
Requirement	<b>/req/observations/classes</b>

SurveyObservations is a specialization of OM\_Observation defined in ISO 19156. (Figure 42)

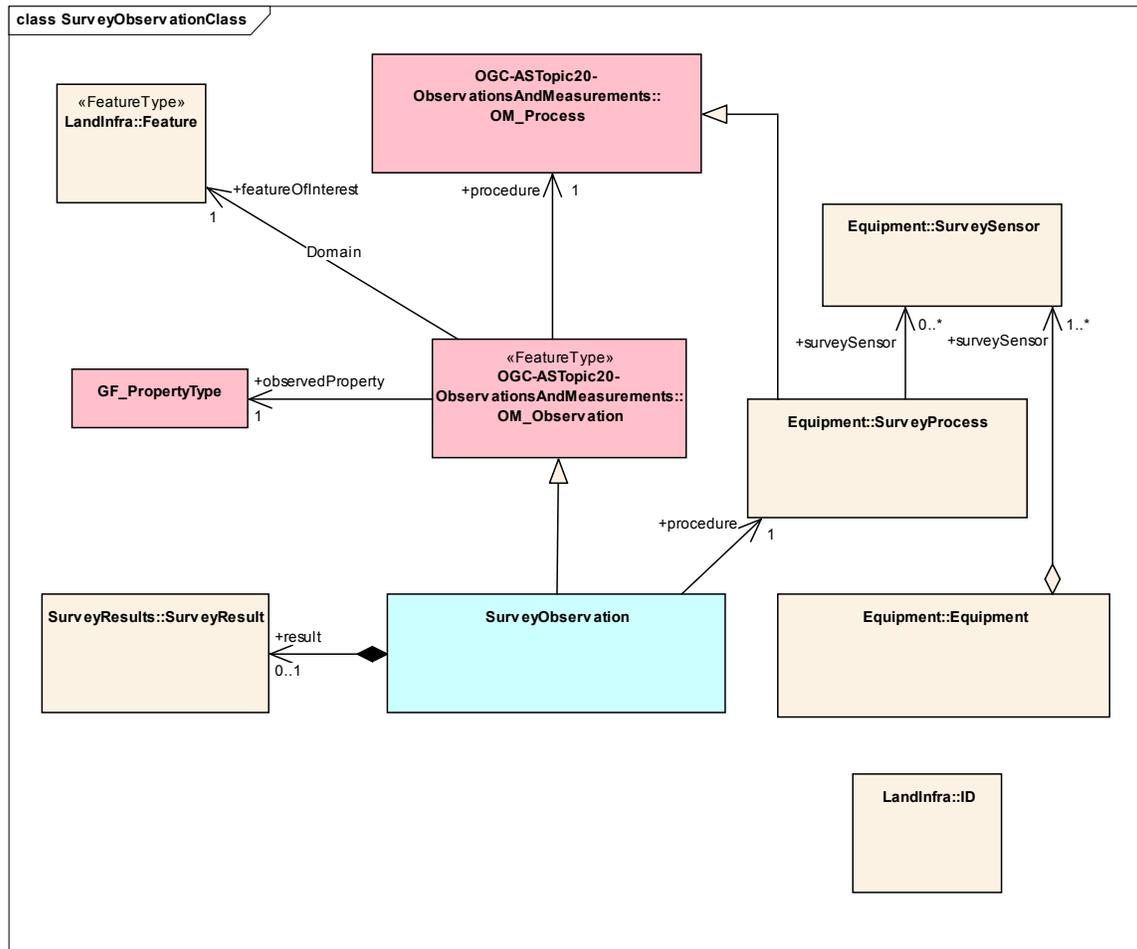


Figure 42. SurveyObservation and OGC Abstract Specification Topic20

A LandInfra dataset may contain any number of Observations to organize all kind of observations based on the location where the measurement has been taken. Observations are organized in different Setups to have the possibility to reprocess the Observations.

The association Domain in Figure 42 shall link the SurveyObservation to the LandInfra Feature that is the subject of the observation and carries the observed property. This feature has the role featureOfInterest with respect to the observation. This feature is the real-world object whose properties are under observation, or is a feature intended to sample the real-world object, as described in Clause 9 of ISO 19156. For SurveyObservations holding a SurveyResult with a Targetpoint (section 7.8.3.4 ) the feature would be most likely a LandInfra SurveyMark.

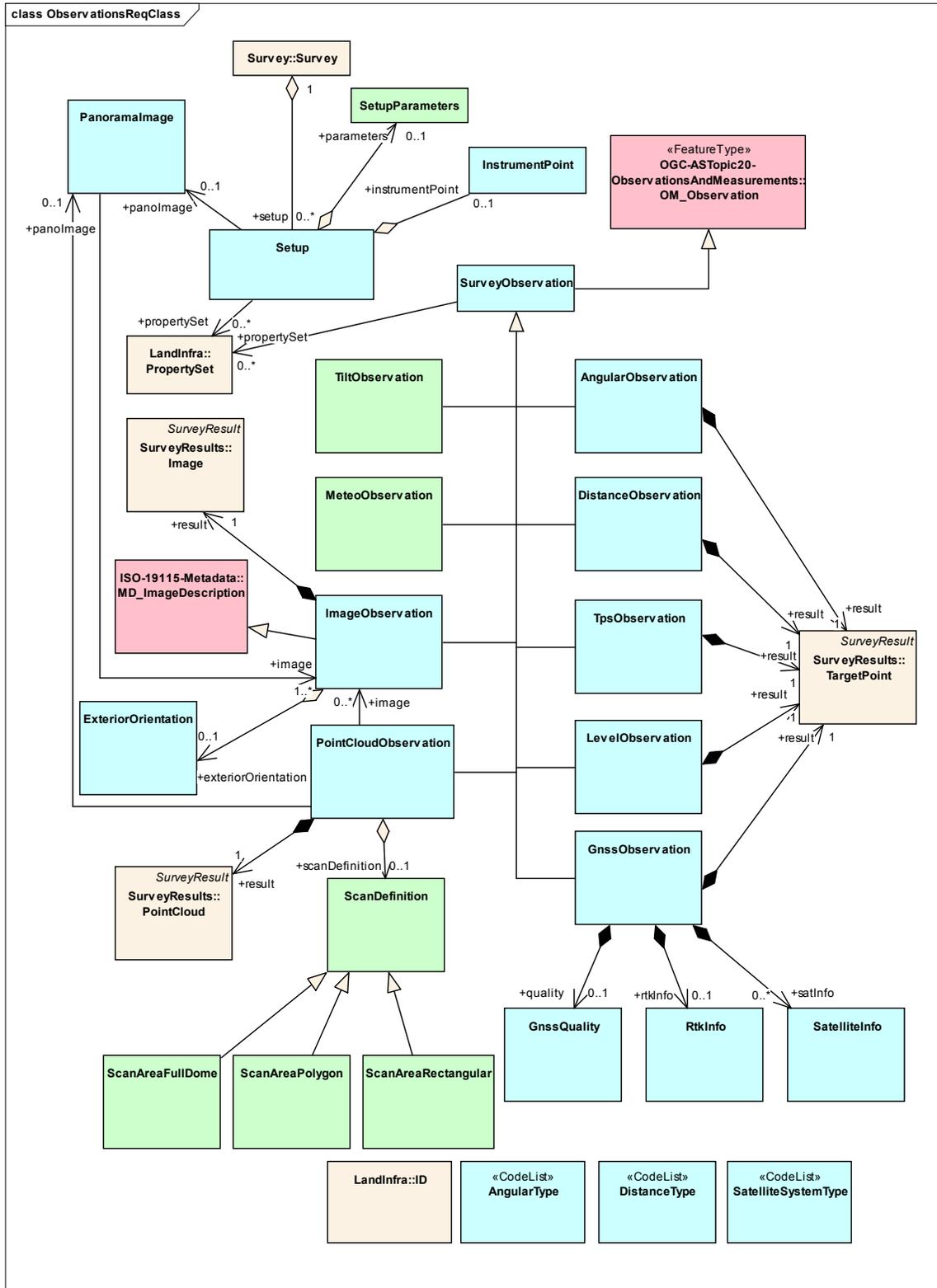


Figure 43. Observations Requirements Class

### 7.8.3 Observations Requirements Class Classes

Requirement	<p><b><a href="#">/req/observations/classes</a></b></p> <p>The Observations Requirements Class Classes shown in blue in Figure 41Figure 43 shall be provided for by the encoding in a manner consistent with the encoding. An encoding shall decide which SurveyObservation types it will support and define appropriate requirements classes accordingly. This then would include any dependent types (example SatelliteSystemType has only to be supported if the encoding will support GNSS Observations).</p>
-------------	---

#### 7.8.3.1 Setup

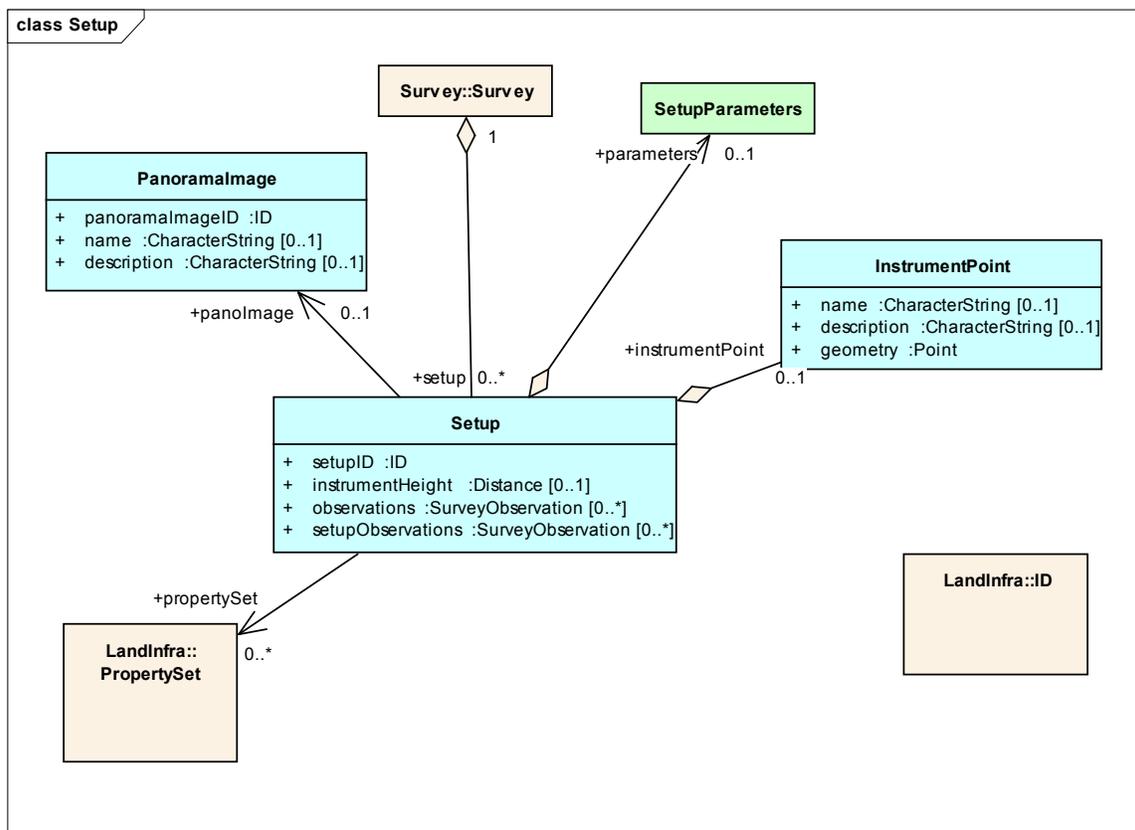


Figure 44. Setup, InstrumentPoint, PanoramaImage and SetupParameters Context Diagram

A LandInfra dataset may contain any number of Setups. Setup describes the place from where different observations have been measured. Grouping observations into different setup(s) allows for reprocessing the raw data at any time. Information about the setup method, instrument height and the accuracy information of the setup application may be included as attributes. Corrections may be applied to the raw observation to acquire the

final locational property of the feature of request. If observations are taken under same conditions (example same prism constant offset) then same corrections are to be applied to the raw observations. This is realized to a specialization of SurveyProcess used for all of the observation in this setups.

Setups are the container for observations and their corrections and have the following attributes:

**setupID:** user defined unique ID used to identify a setup

**instrumentHeight:** vertical distance between the instrument location and the InstrumentPoint. The InstrumentPoint is the geometric location on the ground and most instruments are setup with a vertical shift to this location (e.g. GNSS antenna on a pole).

**observations:** set of all SurveyObservations collected at this setup

**setupObservations:** set of all SurveyObservations collected at this setup in order to determine the location of the InstrumentPoint.

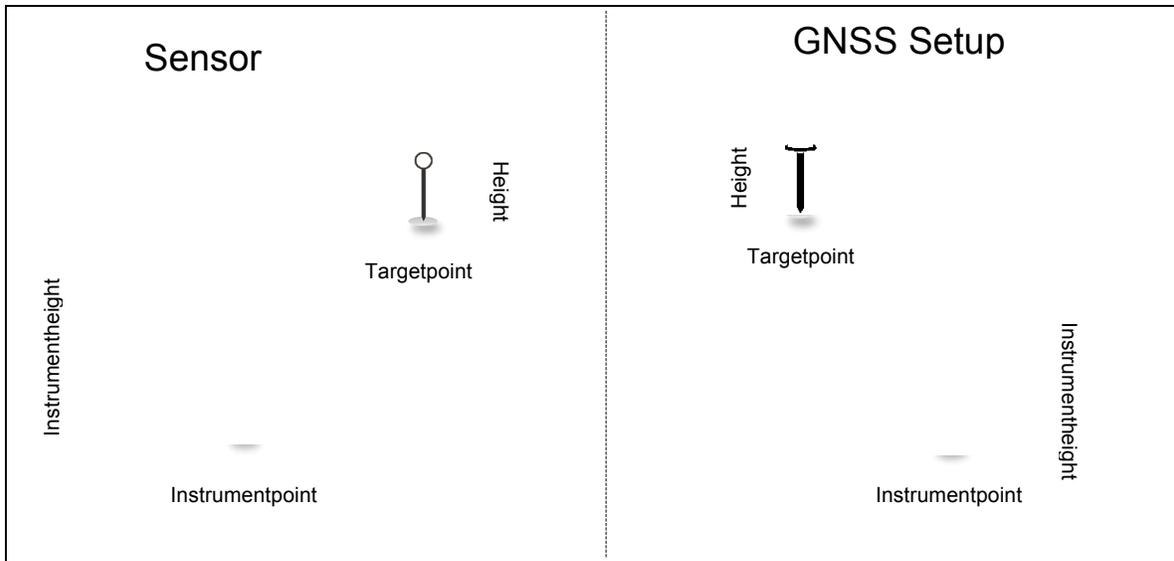
### 7.8.3.2 InstrumentPoint

Knowing the geometric location of the instrument is represented in an InstrumentPoint. For differential GNSS Setups this is the position of the Basestation and for all other Setups this is the location of the used sensor (Figure 45). An InstrumentPoint has the following attributes:

**name:** optional

**description:** optional

**geometry:** geometric location of the instrument point



*Figure 45. InstrumentPoint and Targetpoint*

### 7.8.3.3 PanoramaImage

Panorama images from the setup location is an image which is stitched from single ImageObservations. PanoramaImages allows the visual description of the surroundings of a setup location and are often part of a field book report. A PanoramaImage has the following attributes:

**panoramaImageID:** user defined unique ID used to identify a PanoramaImage

**name:** optional

**description:** optional

### 7.8.3.4 SurveyObservation

There are various observation kinds using today's modern surveying equipment. SurveyObservations are detailed in two separate figures where Figure 46 includes those observation types resulting with a geometric location (TargetPoints) Figure 47 includes those without.



The SurveyObservation is a specialization of OM\_Observation. The derived objects contain the raw measurements from the sensors. Most SurveyObservation types will have the location of the feature of interest (TargetPoint). A generic approach is provided so that simple measurements (e.g. distance from a tape) can be integrated into the model. Multi sensor devices such as survey totalstations (Tps) are represented separately to allow the grouping and sorting of the measurements (example: distance, angular, tilt observations) in an efficient way. SurveyObservation is reading of a property at a specific time from a measurement sensor. A flag describes if the observation was made at InstrumentPoint or TargetPoint position (e.g. an image could be taken at both positions).

SurveyObservation has the following attributes:

**surveyObservationID:** user defined unique ID used to identify a SurveyObservation

**type:** type of the SurveyObservation

**InstrumentPoint:** flag to identify if the observation was taken at a specific setup (InstrumentPoint) or at the location of the feature of interest.

#### 7.8.3.5 AngularObservation

An AngularObservation contains the horizontal or vertical direction from the InstrumentPoint to the TargetPoint.

An AngularObservation has the following attributes:

**angle:** amount of rotation needed to bring one line or plane into coincidence with another

**type:** describes the direction of the rotation

**angularQuality:** quality of the angular observation

#### 7.8.3.6 DistanceObservation

A DistanceObservation contains horizontal / vertical or slope distance from the InstrumentPoint to the TargetPoint.

A DistanceObservation has the following attributes:

**distance:** separation between two points

**type:** describes the direction of the distance measurement

**distanceQuality:** quality of the distance observation

### 7.8.3.7 TPSObservation

A TPSObservation contains angular and distance observation from the InstrumentPoint to the TargetPoint. Additional sensor information as inclination observations maybe also included.

A TPSObservation has the following attributes:

**reflectorHeight:** vertical distance between the reflector and the TargetPoint.

**directFace:** face describes the position of the telescope to the vertical circle reading.  
True = Face1.

**meanFace:** true means readings have been taken in both faces and the observations are the mean value between both readings.

**horizontalAngle:** horizontal angle reading from the InstrumentPoint to the TargetPoint.

**hzAngleQuality:** quality of the horizontal angular observation

**verticalAngle:** vertical angle reading from the telescope axis to the reflector.

**vAngleQuality:** quality of the vertical angular observation

**azimuth:** azimuth from the InstrumentPoint to the TargetPoint

**azQuality:** quality of the azimuth observation

**slopeDistance:** slope distance between the InstrumentPoint and the TargetPoint. A TPS observation also could exist without a distance measurement.

**sDistanceQuality:** quality of the slope distance observation

**horizontalDistance:** horizontal distance between the InstrumentPoint and the TargetPoint. Horizontal distances are corrected if some corrections exist while the slope distance is the raw measurement.

**hzDistanceQuality:** quality of the horizontal distance observation

**inclinationLength:** inclination along the line of sight

**inclLengthQuality:** quality of the inclination along the line of sight

**inclinationCross:** inclination across the line of sight.

**inclCrossQuality:** quality of the inclination across the line of sight

### 7.8.3.8 LevelObservation

A LevelObservation contains elevation difference from the InstrumentPoint to the TargetPoint.

A LevelObservation has the following attribute:

**deltaHeight:** elevation difference between two points

**deltaHeightQuality:** quality of the delta height between the points

### 7.8.3.9 GnssObservation

GNSS observation contains all observations to one or more satellite systems. When using differential GNSS, also the information on the reference stations are included.

A GnssObservation has the following attributes:

**antennaHeight:** vertical distance between the reflector and the TargetPoint.

### 7.8.3.10 RtkInfo

Information about the GNSS reference system used for position calculation.

A RtkInfo has the following attributes:

**networkSolution:** false if single baseline was used – true if a reference network was used to calculate the RTK corrections.

**networkType:** type of RTK network that was used.

**dataFormat:** date format for the transmission of the RTK corrections.

**insideRTKNetwork:** true if the rover position was inside the RTK network.

**mountpoint:** name of the mountpoint to connect to the RTK network.

**numNetworkReferences:** number of reference station for calculation the RTK correction.

**numRtkPositionsUsed:** number of RTK positions that has been calculated to determine the position of the TargetPoint.

#### 7.8.3.11 GnsQuality

GnsQuality contains the Dilution of Precision (DOP) information to specify the effect of satellite geometry on positional measurement precision.

A GnsQuality has the following attributes:

**hDOP:** Horizontal Dilution of Precision (HDOP)

**gDOP:** Geometric Dilution of Precision (GDOP)

**pDOP:** Position (3D) Dilution of Precision (PDOP)

**vDOP:** Vertical Dilution of Precision (VDOP)

**tDOP:** Time Dilution of Precision (TDOP)

#### 7.8.3.12 SatelliteInfo

Information about the GNSS satellite system used for position calculation.

A SatelliteInfo has the following attributes:

**systemType:** Type of satellite system that was used for the position calculation.

**numSatsTracked:** number of satellites that has been tracked from the rover position.

**numSatsUsed:** number of satellites that has been used for the position calculation.

#### 7.8.3.13 ObservationType

This enumeration describes the type of the SurveyObservation.

#### 7.8.3.14 SatelliteSystemType

Enumeration that defines the satellite system (gps, glonass, galileo or beidou) used.

#### **7.8.3.15 DistanceType**

Definition of the orientation of the distance measurement.

#### **7.8.3.16 AngularType**

Definition of the orientation of the angular measurement.

#### **7.8.3.17 GnsNetworkType**

Enumeration for different types of NRTK corrections: a Virtual Reference Station (VRS), Flächen Korrektur Parameter (FKP), individualized Master Auxiliary corrections (iMAX) and a Master Auxiliary Concept (MAC).

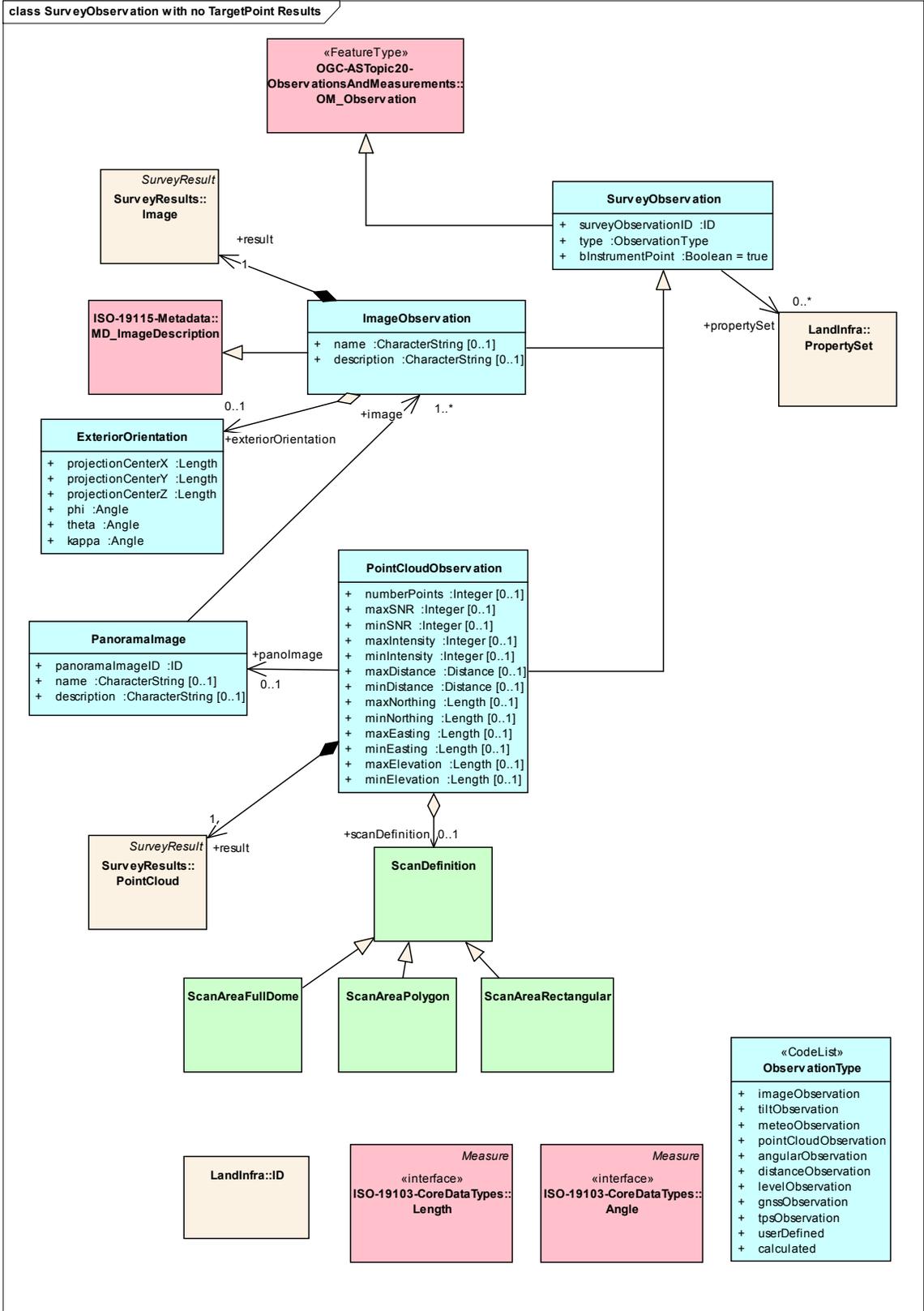


Figure 47. SurveyObservation Classes with no TargetPoint results.

### 7.8.3.18 ImageObservation

An ImageObservation may include an exterior orientation parameter together with the current camera settings to enable the later processing of the images and is a specialization from MD\_ImageDescription defined in ISO 19115.

An ImageObservation has the following attributes:

**name:** optional

**description:** optional

### 7.8.3.19 ExteriorObservation

An ExteriorObservation holds the parameter for the definition of the camera location and its viewing direction when the image was captured.

An ExteriorObservation has the following attributes:

**projectionCenterX:** X coordinate of projection center

**projectionCenterY:** Y coordinate of projection center

**projectionCenterZ:** Z coordinate of projection center

**omega:** rotation about the x-axis, also called roll

**phi:** rotation about the y-axis, also called pitch

**kappa:** rotation about the z-axis, also called yaw

### 7.8.3.20 PointCloudObservation

An PointCloudObservation hold some meta information about the point cloud and the link to the cloud as result. Note that point clouds usually contain a large number of points that have to be managed differently as geometry points in a LandInfra dataset. The dataset may include a link to a panorama image that is used to colorize the point cloud.

An PointCloudObservation has the following attributes:

**numberPoints:** Number of points included in the point cloud

**maxSNR:** quality index of signal to noise ratio

**minSNR:** quality index of signal to noise ratio

**maxIntensity:** maximum intensity value in the cloud

**minIntensity:** minimum intensity value in the cloud

**maxDistance:** maximum distance measure from instrument

**minDistance:** minimum distance measure from instrument

**maxNorthing:** maximum Northing

**minNorthing:** minimum Northing

**maxEasting:** maximum Easting

**minEasting:** minimum Easting

**maxElevation:** maximum Elevation

**minElevation:** minimum Elevation

### 7.8.3.21 ObservationType

This enumeration indicates the different types of observations. If multiple real observations had been done to observe a property of interest this would end in a SurveyResult average. The reduced observation for the average could be calculated from all single observations and would be represented as type calculated.

### 7.8.4 Requirements class: Equipment

Requirements Class	
<b>/req/equipment</b>	
Target type	Encoding of the conceptual model
Name	Equipment
Dependency	<b>/req/survey</b>
Dependency	<a href="http://www.opengis.net/doc/is/om/2.0">http://www.opengis.net/doc/is/om/2.0</a>
Dependency	<a href="urn:iso:is:iso:19130">urn:iso:is:iso:19130</a> clause:7
Requirement	<b>/req/equipment/classes</b>
Requirement	<b>/req/equipment/observation-correction</b>

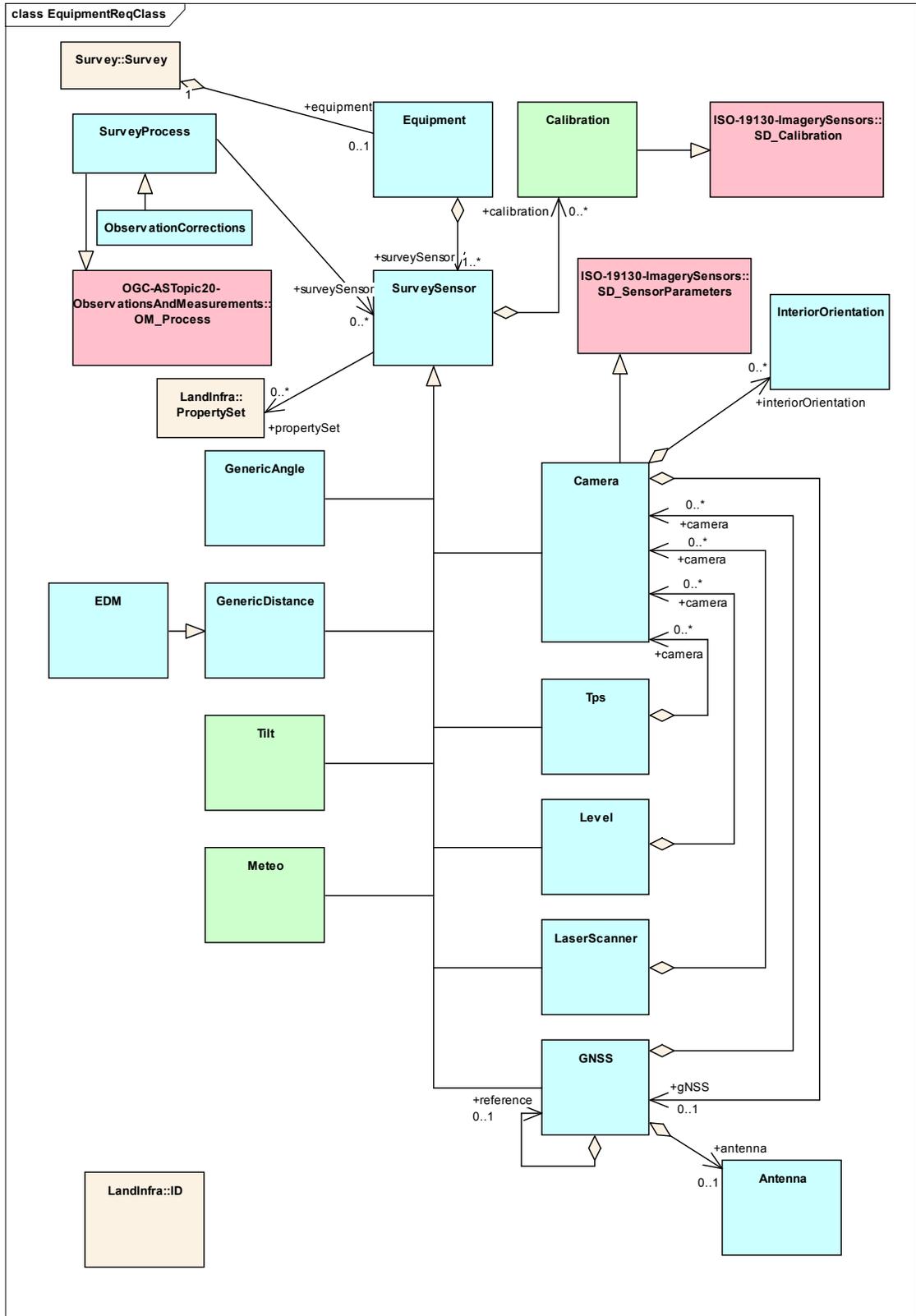


Figure 48. Equipment Requirements Class

A Survey has one Equipment class that contains all information about the sensor(s) used for each observation in a survey. Additionally, attributes contain information about the controlling software that was used to steer the sensors and processes the observations. Calibration of sensors may be required dependent on the sensors used. Some Survey applications requires calibration in the field. The calibration of remote sensing imagery sensors and the validation is covered by ISO 19159-2.

### 7.8.5 Equipment Requirements Class Classes

<b>Requirement</b>	<b><a href="#">/req/equipment/classes</a></b>
	The Equipment Requirements Class Classes shown in blue in Figure 48 shall be provided for by the encoding in a manner consistent with the encoding. An encoding shall decide which SurveySensor types it will support and define appropriate requirements classes accordingly.

#### 7.8.5.1 Equipment

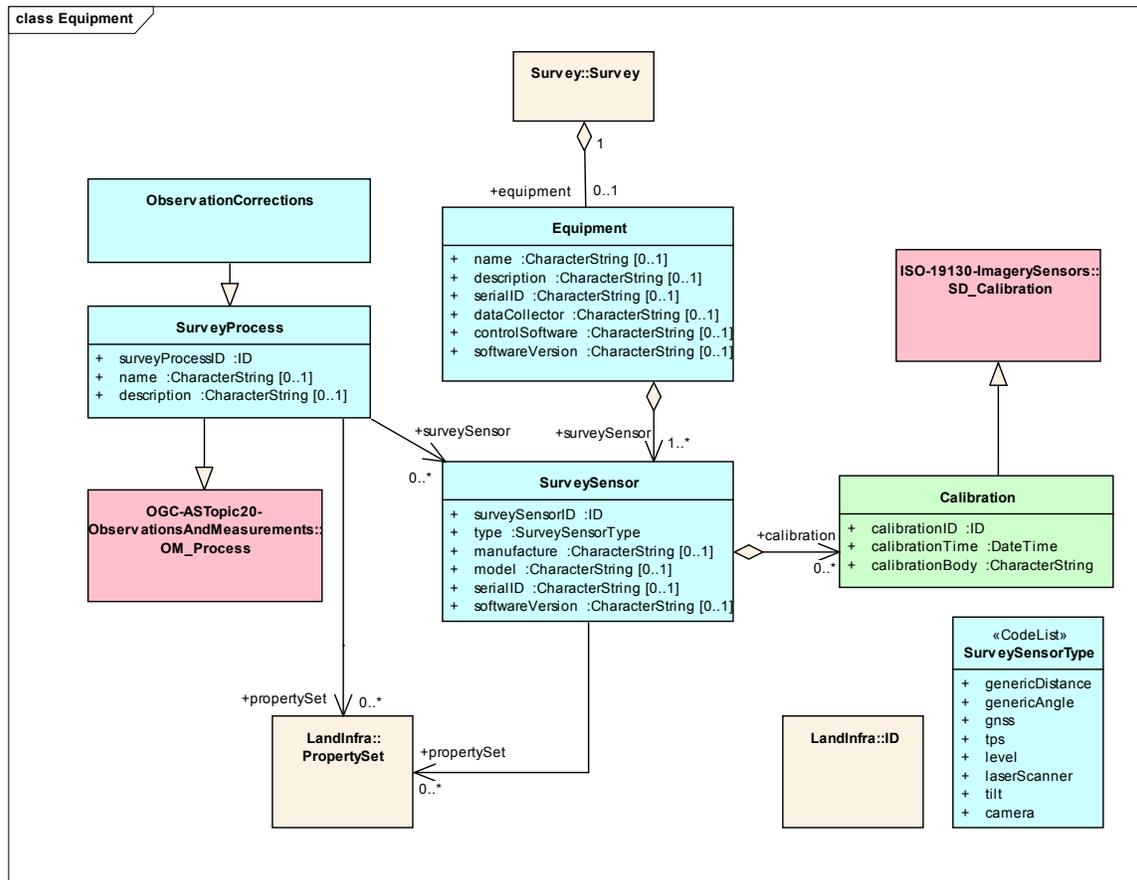


Figure 49. Equipment, SurveySensor and Calibration Context Diagram

A LandInfra dataset may contain any number of Surveys that include one Equipment element.

Equipment is the container for SurveySensors and their Calibrations and has the following attributes:

**name:** optional

**description:** optional

**serialID:** serial number of the hardware where the control software is running.

**dataCollector:** name of the hardware device where the control software is running.

**controlSoftware:** name of the control software.

**softwareVersion:** software version of the control software.

#### 7.8.5.2 SurveyProcess

The purpose of an SurveyProcess is to generate an observation result and it also could contain information about the instrument / sensor that was used to collect measurements. SurveySensor is a specialization of OM\_Process.

SurveyProcess has the following attributes:

**surveyProcessID:** user defined unique ID used to identify a SurveyProcess.

**name:** optional

**description:** optional

#### 7.8.5.3 SurveySensor

SurveySensor contains information about the hardware that is used to collect measurements. Derived SurveySensor types may contain sensor specific attributes.

SurveySensor has the following attributes:

**surveySensorID:** user defined unique ID used to identify a SurveySensor.

**manufacture:** name of the manufacturer.

**model:** sensor model definition.

**serialID:** serial number of the sensor.

**softwareVersion:** software version of the software running on the sensor.

#### 7.8.5.4 Calibration

Each sensor could have factory default calibration values. However, additional field calibration might be necessary. Calibration data contains information about the calibrated values for the measurement sensor and is a specialization of SD\_Calibration. Additional field calibration might be necessary – therefore a sensor could hold one or more calibration records.

Specializations for each sensor specific calibration data and method could be added in the future.

Calibration has the following attributes:

**calibrationID:** user defined unique ID used to identify a Calibration.

**calibrationTime:** date and Time when the sensor was calibrated.

**calibrationBody:** name of the agency or company that executed the calibration.

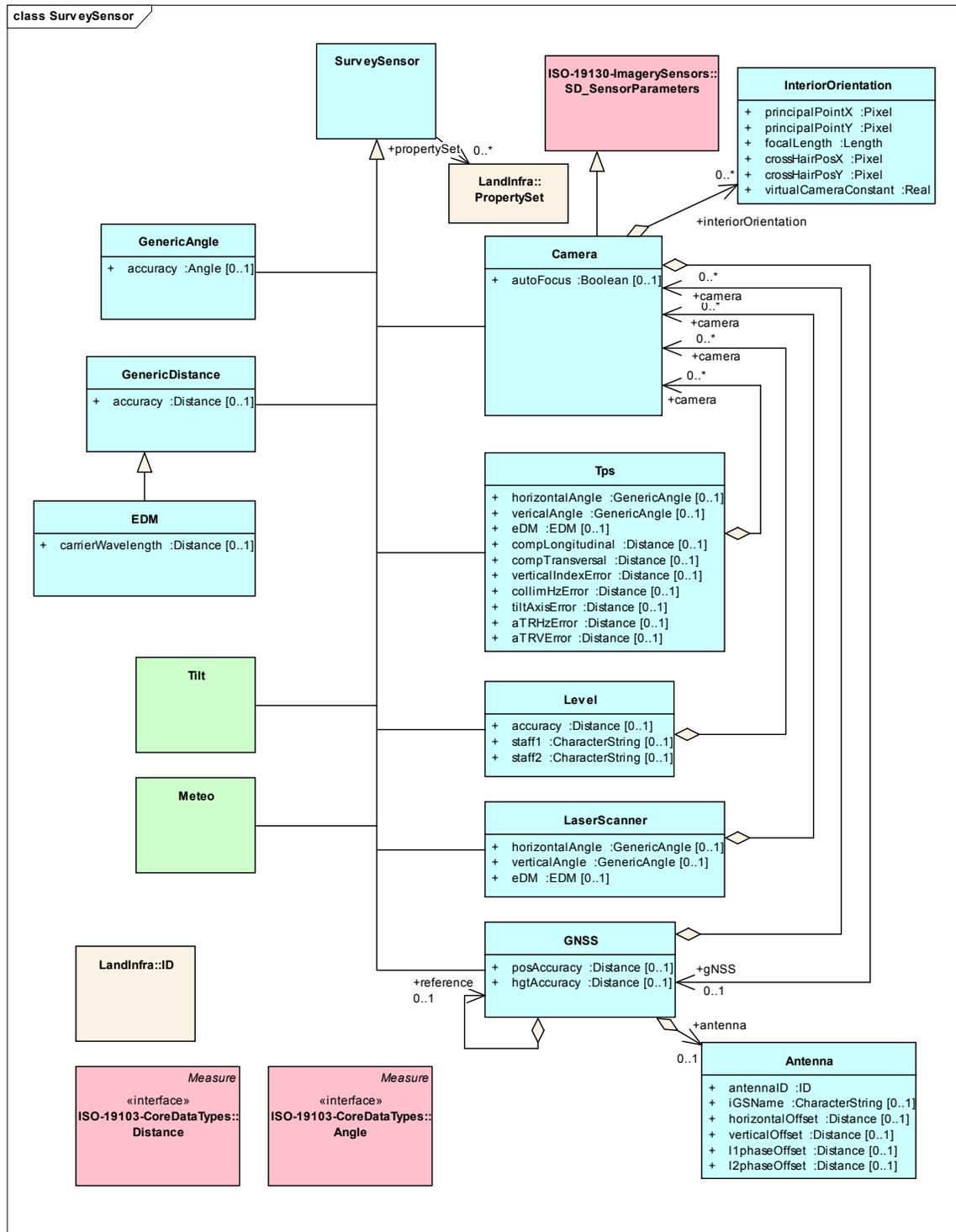


Figure 50. SurveySensor derived Classes.

Figure 50 reflects that a modern measurement sensor could be a multi sensor device. (Camera could have a GNSS chip for example).

#### **7.8.5.5 GenericAngle**

A GenericAngle sensor could be for example a compass.

GenericAngle has the following attributes:

**accuracy:** angle accuracy of the sensor.

#### **7.8.5.6 GenericDistance**

A GenericDistance sensor could be for example a tape.

GenericDistance has the following attributes:

**accuracy:** distance accuracy of the sensor.

#### **7.8.5.7 EDM**

An Electronic Distance Meter (EDM) sensor

EDM has the following attributes:

**carrierWavelength:** wavelength of the EDM.

#### **7.8.5.8 Camera**

A Camera sensor is a specialization of SD\_SensorParameters and hold information about the pixel size and chip window. If the focal length for the camera is always constant, then the interior orientation is always the same. Therefore, this information is attached to the equipment and not to the image observation.

Camera has the following attributes:

**autoFocus:** information if the camera has Autofocus.

#### **7.8.5.9 InteriorOrientation**

This class holds the parameter of the Interior Orientation.

InteriorOrientation has the following attributes:

**principalPointX**: principal point X.

**principalPointY**: principal point Y

**focalLength**: focal length of the lens

**crossHairPosX**: cross hair position X.

**crossHairPosY**: cross hair position Y.

**virtualCameraConstant**: virtual camera constant.

#### 7.8.5.10 Tps

Tps (totalstation) is an electronic theodolite integrated with an electronic distance meter (EDM). Modern totalstations may also contain one or more cameras.

Tps has the following attributes:

**horizontalAngle**: angle accuracy information of the horizontal angle sensor.

**verticalAngle**: angle accuracy information of the vertical angle sensor.

**eDM**: wavelength of the EDM.

**compLongitudinal**: longitudinal index error of the compensator.

**compTransversal**: transversal index error of the compensator.

**verticalIndexError**: vertical index error.

**collimHzError**: horizontal collimation error.

**tiltAxisError**: tilt axis error.

**aTRHzError**: horizontal error of the automatic target recognition system.

**aTRVError**: vertical error of the automatic target recognition system.

#### 7.8.5.11 Level

Level sensors are used to determine the height difference between two locations

Level has the following attributes:

**accuracy**: accuracy of the sensor.

**staff1**: name or ID of the staff that has been used at the TargetPoint.

**staff2**: name or ID of the second staff that has been used at the TargetPoint.

#### 7.8.5.12 LaserScanner

Terrestrial LaserScanner sensors are used measure point clouds with the controlled steering of laser beams followed by a distance measurement at every pointing direction.

LaserScanner has the following attributes:

**horizontalAngle**: angle accuracy information of the horizontal angle sensor.

**verticalAngle**: angle accuracy information of the vertical angle sensor.

**eDM**: wavelength of the EDM.

#### 7.8.5.13 GNSS

GNSS sensors receive the signals from different satellite systems to determine a 3d location of the TargetPoint. External antennas could be also used. If differential GNSS was used GNSS sensor may also contain the sensor information of the reference station. GNSS class will then contain the reference aggregation of a GNNS reference sensor.

GNSS has the following attributes:

**posAccuracy**: Positon accuracy of the sensor.

**hgtAccuracy**: Height accuracy of the sensor.

#### 7.8.5.14 Antenna

GNSS antenna that was used with the sensor

Antenna has the following attributes:

**antennaID**: user defined unique ID used to identify an Antenna.

**iGSName**: antenna Name defined by IGS.

**horizontalOffset**: horizontal offset of measurement reference point.

**verticalOffset**: vertical offset of measurement reference point.

**l1phaseOffset:** offset of L1 phase center.

**l2phaseOffset:** offset of L2 phase center.

### 7.8.5.15 ObservationCorrections

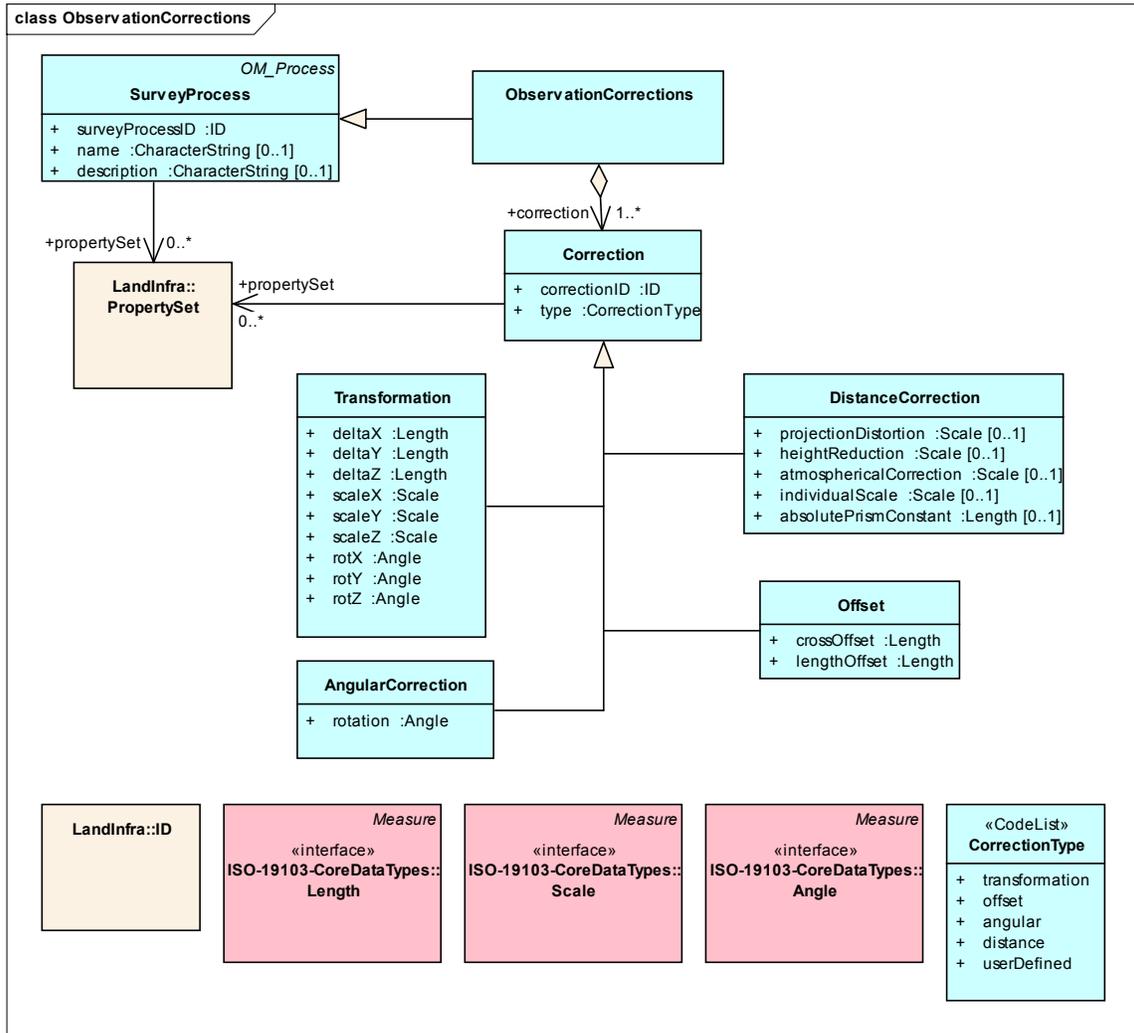


Figure 51. ObservationCorrection Classes.

<b>Requirement</b>	<p><b><i>/req/equipment/observation-correction</i></b></p> <p>The ObservationCorrection Classes in blue in Figure 51 shall be provided for by the encoding in a manner consistent with the encoding. The encoding shall decide which correction types it will support and define appropriate requirements classes accordingly.</p>
--------------------	--

ObservationCorrections is a container for different corrections. Corrections may be identical for several measurements on a setup and therefore collected together in this class.

ObservationCorrections have no attributes:

#### 7.8.5.16 Correction

Correction could be applied to the raw measurements from a sensor. Survey Observations always contain the raw reading from a measurement and individual corrections have to be applied under several conditions.

Correction has the following attributes:

**correctionID:** user defined unique ID used to identify a Correction.

**type:** Type of the correction.

The following sections describes the different correction types.

#### 7.8.5.17 Correction with a Transformation

A Transformation is used to get the TargetPoint in the Target CRS if the Instrument Point is not in the Target CRS.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix} + \begin{pmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & m_z \end{pmatrix} \mathbf{R}(\omega_1, \omega_2, \omega_3) \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Transformation has the following attributes:

**deltaX:** Translation of X. ( $X_0$ )

**deltaY:** Translation of Y. ( $Y_0$ )

**deltaZ:** Translation of Z. ( $Z_0$ )

**scaleX:** scale of X. ( $m_x$ )

**scaleY:** scale of Y. ( $m_y$ )

**scaleZ:** scale of Z. ( $m_z$ )

**rotX:** rotation X axis. ( $\omega_1$ )

**rotY:** rotation Y axis. ( $\omega_2$ )

**rotZ:** rotation Z axis. ( $\omega_3$ )

#### 7.8.5.18 AngularCorrection

AngularCorrection is a rotation that is applied to angular measurement. An example would be the correction from magnetic north to true north.

AngularCorrection has the following attributes:

**rotation:** angular value

#### 7.8.5.19 DistanceCorrection

DistanceCorrection is a scaled or fixed length that is applied to the raw distance measurement.

DistanceCorrection has the following attributes:

**projectionDistortion:** The magnitude of the projection distortion is in accordance with the projection system used in a particular country, for which official tables are generally available.

**heightReduction:** reduction to reference ellipsoid

**atmosphericalCorrection:** value for the correction of atmospheric effects

**individualScale:** user entered scale factor

**absolutePrismConstant:** prism constant

#### 7.8.5.20 Offset

Offset could be applied to the distance measurements along or across the direction from the InstrumentPoint to the TargetPoint.

Offset has the following attributes:

**crossOffset:** offset across the direction

**lengthOffset:** offset along the direction

#### **7.8.5.21 SurveySensorType**

This enumeration the type of sensor used in the equipment.

### 7.8.6 Requirements class: SurveyResults

Requirements Class	
<b>/req/survey-results</b>	
Target type	Encoding of the conceptual model
Name	SurveyResults
Dependency	<b>/req/survey</b>
Dependency	<a href="http://www.opengis.net/doc/is/om/2.0">http://www.opengis.net/doc/is/om/2.0</a>
Requirement	<b>/req/survey-results/classes</b>
Requirement	<b>/req/survey-results/topic-20</b>

A LandInfra dataset may contain any number of SurveyResults that contains the estimate of the value(s) of a geometry or property of the feature of interest. The results could be linked to sample features – this would enable a later reprocessing of all observation to determine how the results have been determined.

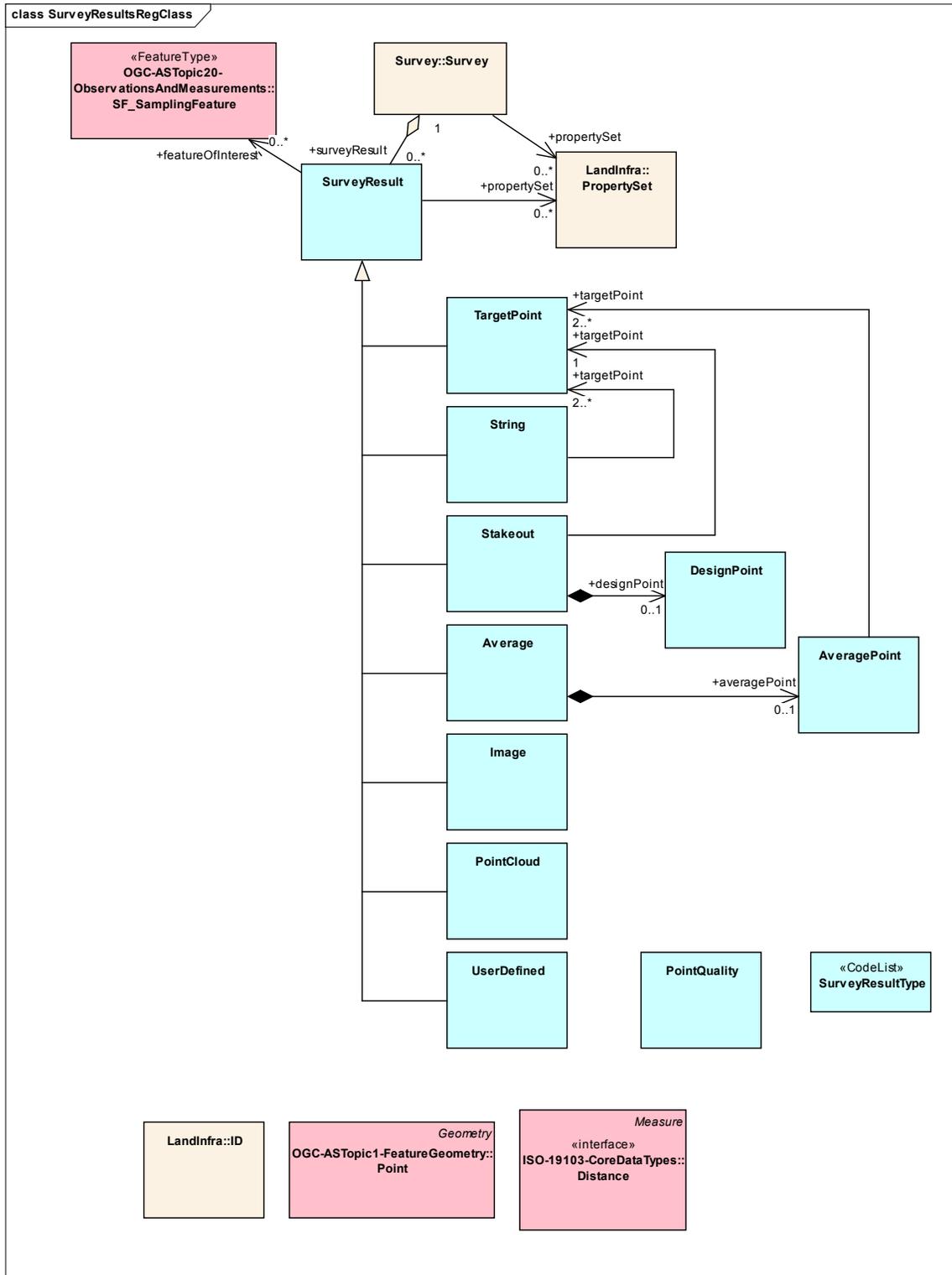


Figure 52. SurveyResults Requirements Class

### 7.8.7 SurveyResults Requirements Class Classes

<b>Requirement</b>	<b><a href="#">/req/survey-results/classes</a></b>  The SurveyResults Requirements Class Classes shown in blue in Figure 52 shall be provided for by the encoding in a manner consistent with the encoding. An encoding shall decide which SurveyResult types it will support and define appropriate requirements classes accordingly.
--------------------	--

### 7.8.7.1 SurveyResults

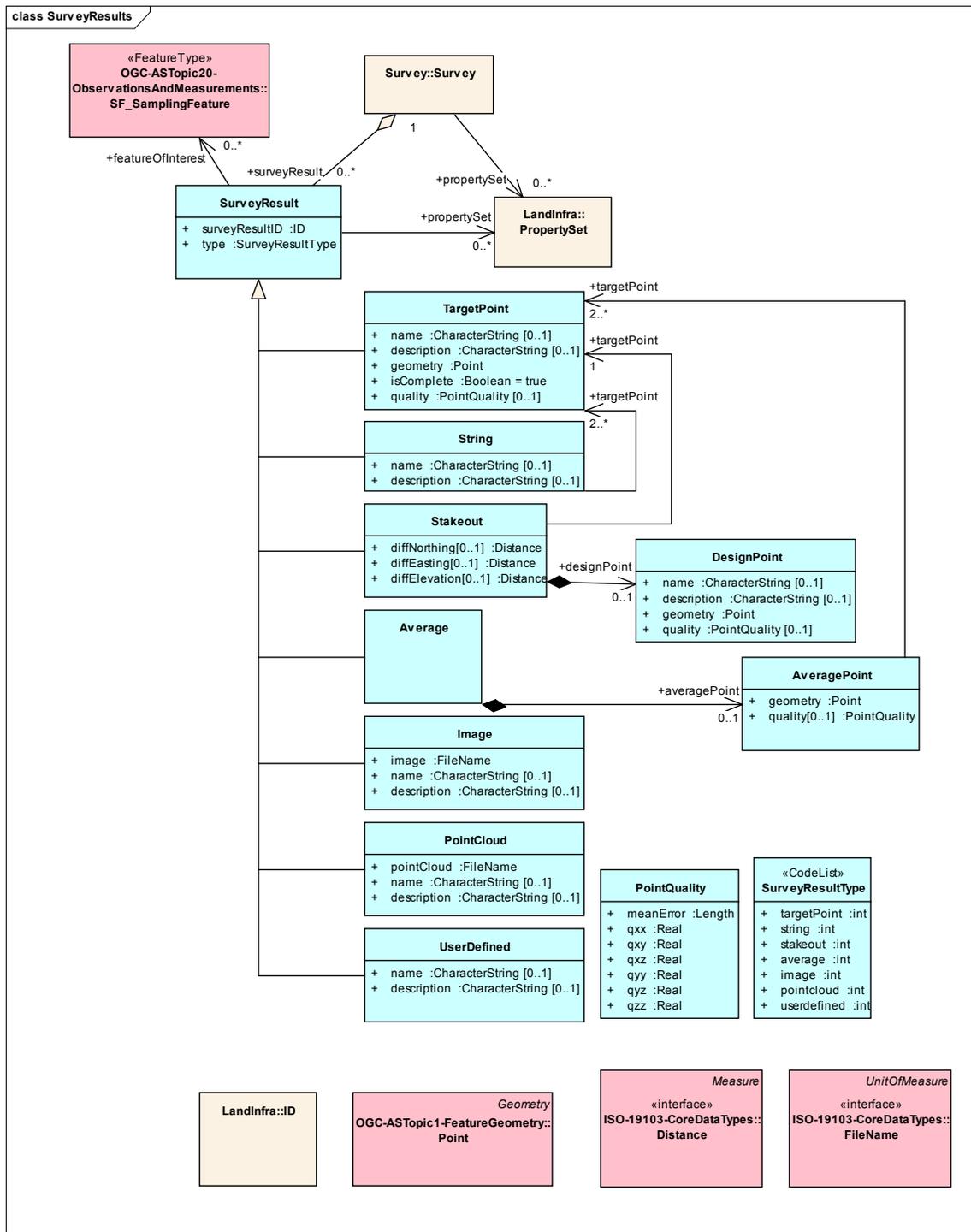


Figure 53. SurveyResults and derived Classes

SurveyResults is an estimate of the value of a geometry or property of the feature of interest. If the result contains any geometry, then this would be represented in the

TargetPoint. SurveyResults will have a link to SF\_SamplingFeature to realize the possibility what process and observation had been used for their acquisition (

OGC Abstract Specification Topic 20 – Observation and Measurements).

SurveyResults have the following attribute:

**surveyResultID:** user defined unique ID used to identify a SurveyResults

#### 7.8.7.2 TargetPoint

TargetPoint is the value of a geometry of the feature of interest that has been estimated from the raw observations.

TargetPoint has the following attributes:

**name:** optional

**description:** optional

**geometry:** geometric location of the target point.

**isComplete:** TargetPoint has not a zero-dimensional geometric representation (a single representative Point) (isComplete = TRUE)

**quality:** quality information of the TargetPoint.

#### 7.8.7.3 String

If a string were surveyed, then the class has two or more links to target points. There is no link to an observation in the sampling feature for this class.

String has no attributes – the sampling feature will link to the related feature geometry element that is represented in the LandInfra dataset.

String has the following attributes:

**name:** optional

**description:** optional

#### 7.8.7.4 Stakeout

Stakeout results contains the differences between the design and the resulted TargetPoint. The design point geometry is hold in the DesignPoint.

Stakeout has the following attributes:

**diffNorthing:** stakeout differences in Northing.

**diffEasting:** stakeout differences in Easting.

**diffElevation:** stakeout differences in Elevation.

#### 7.8.7.5 DesignPoint

Design point that had been staked out.

DesignPoint has the following attributes:

**name:** optional

**description:** optional

**geometry:** geometric location of the design point.

**isComplete:** TargetPoint has not a zero-dimensional geometric representation (a single representative Point) (isComplete = TRUE)

**quality:** quality information of the design point.

#### 7.8.7.6 Average

Average represents multiple measurements of the same target location. Averaging may be required in order to determine the geometry of the feature for when observations are made at different times. Related observation to an average will be a computed reduced observation.

#### 7.8.7.7 AveragePoint

If the observations result in a TargetPoint, the AveragePoint will hold the average geometry of all single TargetPoints.

AveragePoint has the following attributes:

**geometry:** geometric location of the design point.

**isComplete:** AveragePoint has not a zero-dimensional geometric representation (a single representative Point) (isComplete = TRUE)

**quality:** quality information of the calculated average point.

#### 7.8.7.8 Image

Image is the result of an ImageObservation.

Image has the following attributes:

**name:** optional

**description:** optional

**image:** URL to the image file.

#### 7.8.7.9 PointCloud

PointCloud is the result of a PointCloudObservation.

PointCloud has the following attributes:

**name:** optional

**description:** optional

**pointCloud:** URL to the pointcloud file.

#### 7.8.7.10 UserDefined

**name:** optional

**description:** optional

#### 7.8.7.11 PointQuality

PointQuality include the information of the reached quality with the processed observations.

PointQuality has the following attributes:

**meanError:** mean error of unit weight of the coordinates.

**qxx:** X component of variance covariance matrix (q matrix).

**qxy:** X-Y component of variance covariance matrix (q matrix).

**qxz:** X-Z component of variance covariance matrix (q matrix).

**qyy:** Y component of variance covariance matrix (q matrix).

**qyz:** Y-Z component of variance covariance matrix (q matrix).

**qzz:** Z component of variance covariance matrix (q matrix).

### 7.8.8 OGC Abstract Specification Topic 20 – Observation and Measurements

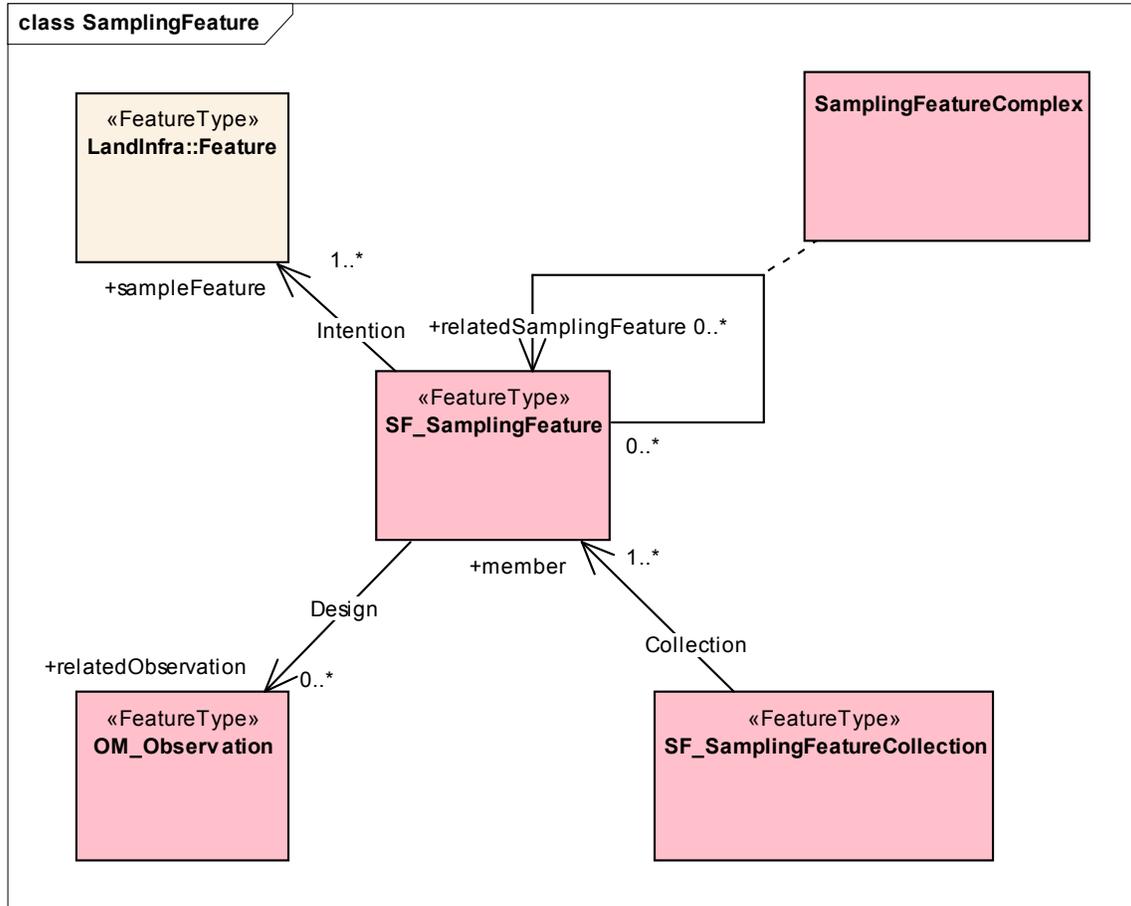


Figure 54. OGC Abstract Specification Topic 20 – Sampling Feature

Having navigable associations from the survey results to the related observations is needed for many survey applications (for example monitoring purposes, multiple measurements are required at different times).

A sampling feature is established in order to make observations concerning some domain feature. The association Intention shall link the SF\_SamplingFeature to the feature which the sampling feature was designed to sample. The target of this association has the role sampledFeature with respect to the sampling feature, and shall not be a sampling feature or observation. It is usually a real-world LandInfra feature.

<b>Requirement</b>	<b><a href="#">/req/land-infra/topic-20</a></b>  If a Land and Infrastructure encoding supports survey results subjects, then the encoding shall support the SF_SamplingFeature and related Classes specified in the OGC Abstract Specification Topic 20, Observation and Measurements shown in Figure 54 that are appropriate to the supported subject area(s).
--------------------	--

## 7.9 Requirements class: LandFeature

Requirements Class	
<b>/req/land-feature</b>	
Target type	Encoding of the conceptual model
Name	Land Feature
Dependency	<b>/req/land-infra</b>
Requirement	<b>/req/land-feature/classes</b>
Requirement	<b>/req/land-feature/alignment</b>

Features of the land, such as naturally occurring water features and vegetation are specified as LandFeatures. Also included are models of the land surface and subsurface layers. Man-made interventions to the land such as the construction of an embankment or the planting of landscape material are considered to be part of Site Facilities in the Facility Requirements Class.

<b>Requirement</b>	<b>/req/land-feature/classes</b>
	The LandFeature Requirements Class Classes shown in blue in Figure 55 shall be provided for by the encoding in a manner consistent with the encoding.

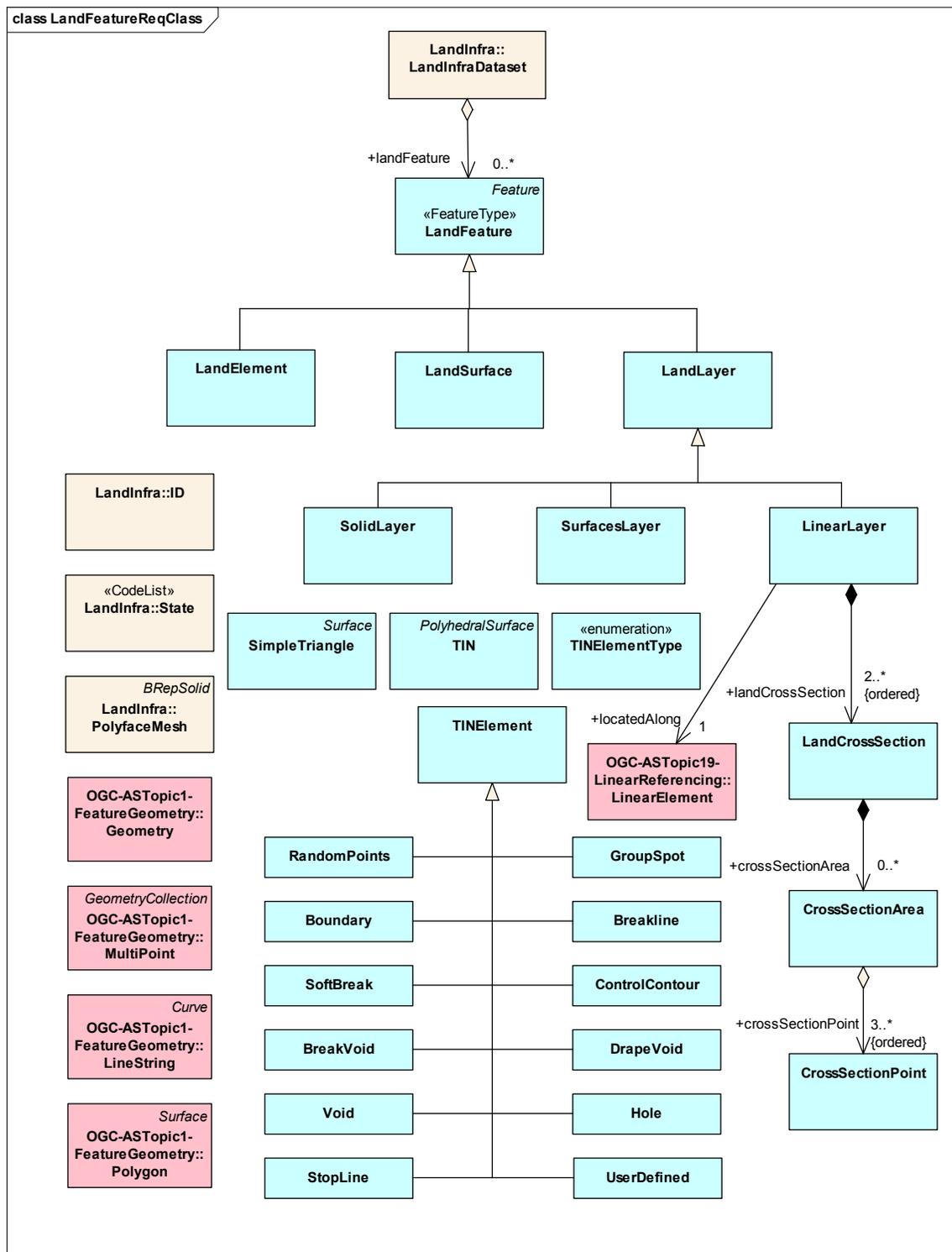


Figure 55. LandFeature Requirements Class



As a subtype of Feature, LandFeature inherits attributes featureID, name, description, spatialRepresentation, linearlyReferencedLocation, property, and propertySet. All LandFeatures also have the following attributes:

**landFeatureID:** identifier for a LandFeature, unique within the LandInfra dataset or globally unique with the inclusion of an ID.scope value

**state:** existing or proposed

## 7.9.2 Land Element

LandElements represent topographical features of the land. These are typically naturally occurring (rivers, vegetation), although they may be the result of previous man-made projects (embankments or landscape plantings). LandElement has the following attributes:

**elementID:** unique identifier for the LandElement, unique within the LandInfra dataset or globally unique with the inclusion of an ID.scope value

**elementType:** the type of LandElement

**material:** the optional material of the element

### 7.9.2.1 Land Element Type

LandElementType defines the type of LandElement, as shown in Figure 56, e.g., waterbody, vegetation, or landForm. LandElementType is defined as a CodeList in accordance with ISO 19103. The inclusion of the optional tagged value codeList: URI choice, is in accordance with ISO 19103. If present, then the only valid values are those from an external authority referenced code list identified by the URI value. If not, the other CodeListed values are valid along with possible additional, user defined values. However, these additional values should not replace an existing code by changing the name or definition, or have the same definition as an existing value.

## 7.9.3 Land Surface

The surface (terrain) of an area of land is specified by a LandSurface. LandSurfaces can also be used to specify the boundary between two subterranean layers. The spatial representation of a LandSurface is specified by a TIN triangulated surface (see 7.9.8). LandSurface has the following attributes:

**spatialRepresentation:** TIN

**material:** optional, the material which lies directly beneath this LandSurface

### 7.9.4 Land Layer

Below the LandSurface, the land comprises layers of differing materials. LandLayer is the abstract class for specifying these material layers. They can be defined in three ways: by a top and bottom horizontal surface, by a series of vertical cross section end areas, or as a 3D Polyface Mesh solid.

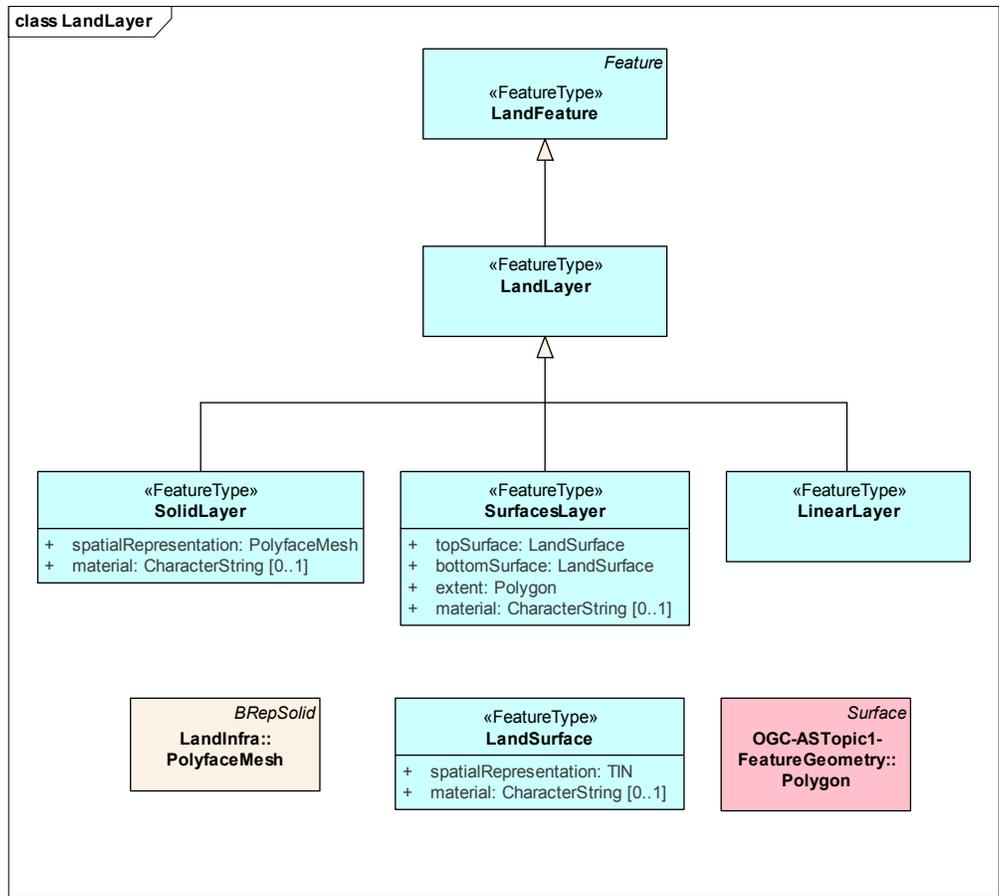


Figure 57. LandLayer context diagram

### 7.9.5 Solid Layer

SolidLayer is used to define a LandLayer using a solid geometry (PolyfaceMesh) to define the spatial representation of the layer. SolidLayer has the following attributes:

**spatialRepresentation:** PolyfaceMesh

**material:** optional, the SolidLayer's material

### 7.9.6 Surfaces Layer

For a SurfacesLayer type of LandLayer, a top and bottom LandSurface is specified. A resultant solid shape is implied between these two layers, within the boundary of the SurfacesLayer's extent, having vertical sides where the surfaces do not connect.

SurfacesLayer has the following attributes:

**topSurface:** a LandSurface defining the top of the SurfacesLayer

**bottomSurface:** a LandSurface defining the bottom of the SurfacesLayer

**extent:** the spatial limits of the layer

**material:** optional, the SurfacesLayer's material

The extent is defined as a Polygon so it can have both an exterior and any number of interior boundaries, the latter defining holes in the layer.

If there are voids or holes in the surfaces, the type of void will dictate how this is manifested in the resultant layer (see 7.9.8.3, TIN Element Type).

If a material is specified, it takes precedence over the material specified for the top surface.

## 7.9.7 Linear Layer

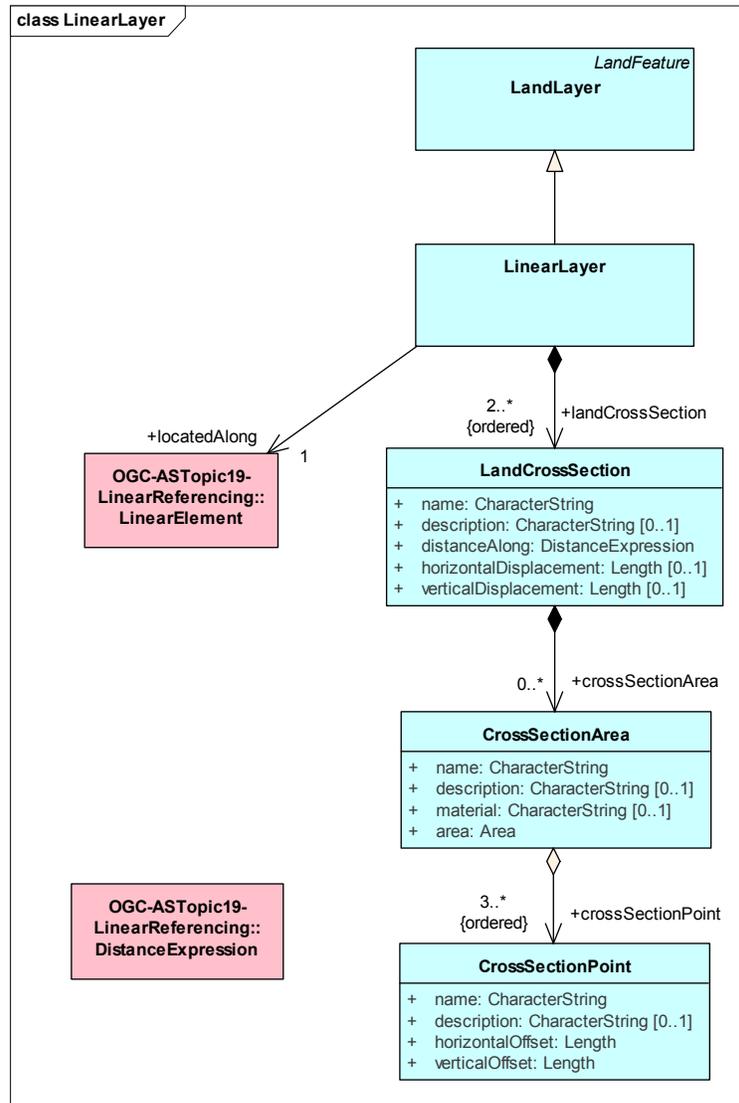


Figure 58. LinearLayer context diagram

A LinearLayer is a layer defined by 2D cross sectional areas along a linear element. LandSurface has the following attributes:

**landCrossSection:** two or more LandCrossSections, ordered along the linear element

**locatedAlong:** the linear element along which the LandCrossSections are located

<b>Requirement</b>	<b><a href="#">/req/land-feature/alignment</a></b>  If an encoding allows the linear element used for a LandLayer to be an Alignment, then that encoding shall support the Alignment Requirements Class.
--------------------	--

### 7.9.7.1 Land Cross Section

A LandCrossSection describes how a LandLayer looks (or will look), in a 2D cross section view, at some specific location along a linear element line of reference.

The two LandCrossSection axes represent the horizontal and vertical offsets for CrossSectionPoints. The axis origins represent a hook point for the LandCrossSection which is used to specify how the LandCrossSection is attached along a linear element. The horizontal offset coordinates measure distances right (positive) or left (negative) and the vertical offset coordinates measure the distance up (positive) or down (negative) from the linear element at the location specified for that LandCrossSection. The LandCrossSection location is specified as a linear element distance along value.

LandCrossSection has the following attributes:

**name:** user defined name used to identify the LandCrossSection, unique within a LandFeature

**description:** optional explanation of the LandCrossSection

**distanceAlong:** the distance measured along the linear element to get to the LandCrossSection location

**horizontalDisplacement:** optional lateral distance measured left (negative) or right (positive) perpendicular from the linear element at the distance along location to the LandCrossSection origin

**verticalDisplacement:** optional vertical distance measured up (positive) or down (negative) from the linear element to the LandCrossSection origin

**crossSectionArea:** any number of CrossSectionAreas

### 7.9.7.2 Cross Section Area

CrossSectionAreas are logical parts of a LandCrossSection, represented in 2D cross section view. Portions of the LandCrossSection that are separated into CrossSectionAreas are normally based on material considerations. Examples of CrossSectionAreas include areas of a particular kind of material.

CrossSectionArea has the following attributes:

- name:** user defined name used to identify the CrossSectionArea, unique within a LandCrossSection
- description:** optional explanation of the CrossSectionArea
- material:** the CrossSectionArea's optional material which can be general such as cut or fill or more specific such as a type of soil or rock
- area:** the cross sectional area of the CrossSectionArea
- crossSectionPoint:** three or more CrossSectionPoints which define the geometry of the CrossSectionArea in "connect the points" order

### 7.9.7.3 Cross Section Point

CrossSectionPoints are 2D points contained within a LandCrossSection view which are used to define the geometry of CrossSectionAreas. A local Cartesian coordinate system is established for the LandCrossSection with origin (0,0) being the location of the LandCrossSection hook point. CrossSectionPoints have coordinate values representing the known horizontal and vertical offset displacements from this origin.

CrossSectionPoint has the following attributes:

- name:** user defined name used to identify the point, unique within a LandCrossSection
- description:** optional explanation of what the point represents
- horizontalOffset:** distance measured horizontally right (positive) or left (negative) from the LandCrossSection hook point
- verticalOffset:** distance measured vertically up (positive) or down (negative) from the LandCrossSection hook point

Care should be taken in selecting CrossSectionPoint names. LandCrossSection Point names shall be unique within a single LandCrossSection. However, it is advisable to use consistent naming across all of the LandCrossSections used for a LandFeature, e.g., "rightToeOfSlope". These similarly named CrossSectionPoints can be connected up along the LandFeature, resulting in a string lines (long section or profile) having the same name as the contributing CrossSectionPoints. The "rightToeOfSlope"

CrossSectionPoints in consecutive LandCrossSections along the LandFeature are thus connected together to create a rightToeOfSlope string line feature in 3D.

### 7.9.8 TIN

Triangulated Irregular Network (TIN) is a subtype of PolyhedralSurface composed of contiguous triangle surface (SimpleTriangle) patches connected along their common boundary linestrings. This differs from PolyhedralSurface in the restriction on the types of surface patches acceptable and the addition of TIN elements which constrain the triangulation. TIN generation uses the Delaunay algorithm, or a similar implementation-defined algorithm, complemented with consideration for breaklines, soft breaks, control contours, break voids, drape voids, voids, holes, stop lines and maximum length of triangle sides.

The use of TIN specified in this standard is an extension of the OGC Abstract Specification Topic 1 Geometry TIN. TIN as defined in Topic 1 includes control points, stop lines, breaklines and maximum triangle side length as well as the resultant triangles. However, the triangle and stop line implementations are problematic for all but the smallest of TINs. Triangles are unnecessarily too verbose, being implemented as ringed polygons with four vertices. Stop lines can create voids but, as linestrings, they are inefficient for specifying large void areas involving many triangles.

The TIN specified in this LandInfra Standard supports significantly larger surfaces with simpler triangles and additional, more reasonably defined TIN elements, such as different types of voids. This approach has been encoded with GML in GML 3.3 and in SQL with SQL/MM Part 3: Spatial. The TIN uses the Delaunay algorithm which requires that for each triangle in the surface, a circle passing through its vertexes does not contain, in its interior, the vertex of any other triangle. This is complemented with additional, optional, TINElement constraints.

A TIN example is included as Annex E, TIN Example from the initial SQL/MM proposal [16].

A TIN has the following attributes:

**patch:** one or more SimpleTriangles

**element:** zero or more TINElements

**maxSideLength:** the maximum length of a side of a SimpleTriangle contained in the surface

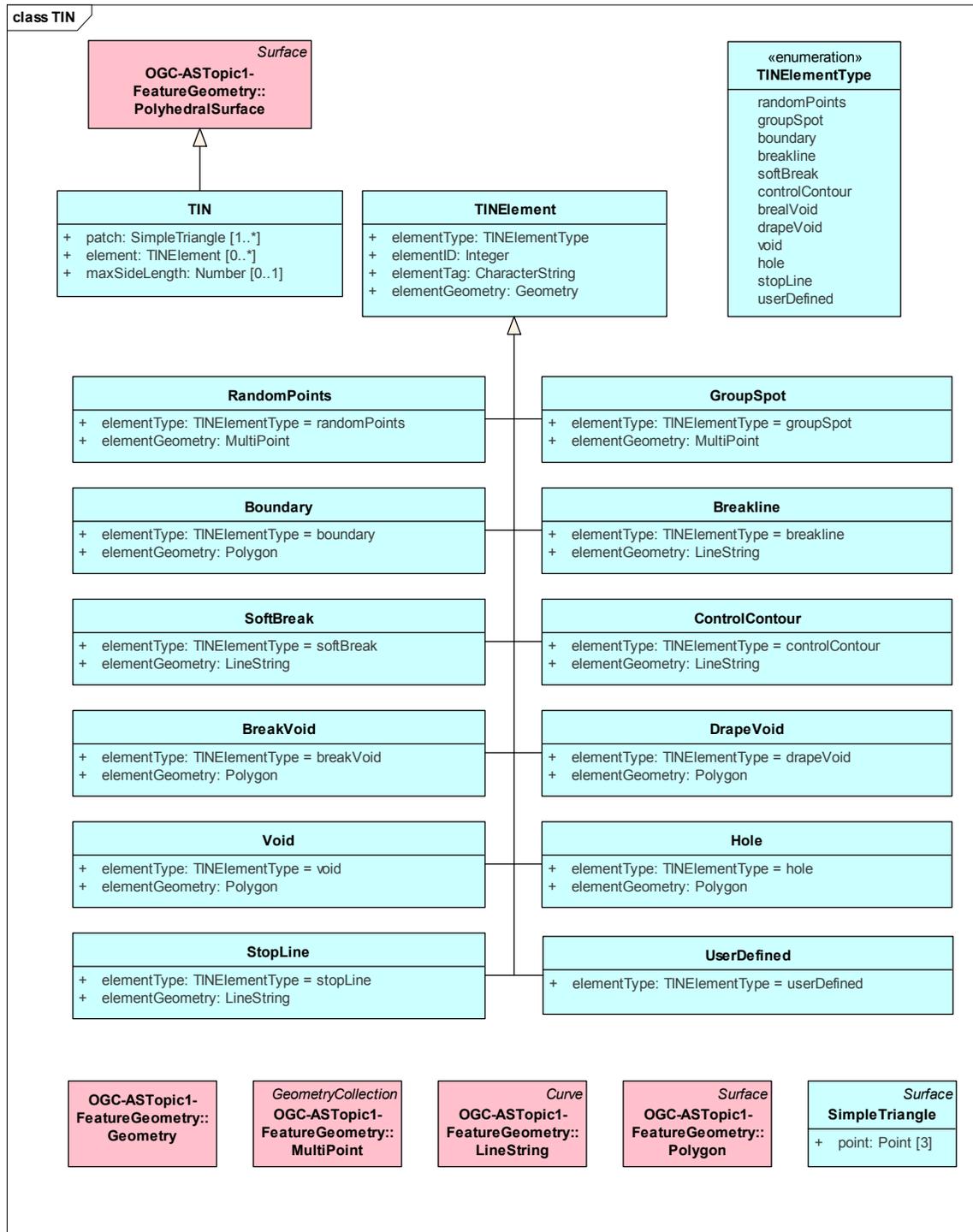


Figure 59. TIN context diagram

### 7.9.8.1 Simple Triangle

SimpleTriangle is a surface defined by three points. This is a simplification of the OGC Abstract Specification Topic 1 Geometry Polygon which requires intermediary ring(s). SimpleTriangle has the following attributes:

**point:** three Points which, when connected, form a clockwise triangle; the first point is repeated as the last, closing point

### 7.9.8.2 TIN Element

TINElements are used to specify the information used to construct a TIN surface. Element types include random points, group spot, boundary, breakline, soft break, control contour, break void, drape void, void, hole, stop line and user defined element types.

All TINElements have the following attributes:

**elementType:** specifies the type of TINElement. Allowable values are specified by TINElementType

**elementID:** optional ID value to identify the TINElement unique within the TIN

**elementTag:** optional alphanumeric identifier for the TINElement unique within the TIN

**elementGeometry:** specifies the geometry of the TINElement. The allowable geometry type is dependent upon the value of the TINElementType:

- random points and group spot TINElements have a 3D MultiPoint geometry
- boundary, break void and void TINElements have a 3D Polygon geometry
- drape void and hole TINElements have a 2D or 3D Polygon geometry
- break line, soft break and control contour TINElements have a 3D LineString geometry
- stop line TINElements have a 2D or 3D LineString geometry
- for a user-defined TINElement, the choice of geometry is user-defined.

### 7.9.8.3 TIN Element Type

TINElementType defines the type of TINElement. Allowable values include:

**randomPoints**

**groupSpot**

**boundary**

**breakline**

**softBreak**

**controlContour**

**breakVoid**

**drapeVoid**

**void**

**hole**

**stopLine**

**userDefined**

A TINElement of type RandomPoints represents points on the surface of known elevation from which triangles can be generated or which represent linestring points contained in other TINElement geometries.

A TINElement of type GroupSpot represents a collection of related points on the surface of known elevation from which triangles can be generated or which represent linestring points contained in other TINElement geometries.

A TINElement of type Boundary is used to specify the boundary of the TIN surface. As a constraint applied to a TIN surface after the initial Delaunay triangulation, application of the boundary causes the surface to be clipped to the Boundary's Polygon geometry value. This may result in the elimination or addition of points contained in the initial random points or group spots TINElements. This may also result in localized re-triangulation at the boundary. Support for interior boundaries is implementation specific.

A TINElement of type Breakline is used to represent a local ridge or depression in the TIN surface. When a breakline is specified for a TIN surface, simple triangle patches

must be adjusted so that no triangle is crossed by the breakline. Part or all of the breakline becomes an edge of two or more triangles. The elevation along the breakline takes precedence over the elevation of the original triangulated surface for the entire length of the breakline.

A TINElement of type SoftBreak behaves as a 'breakline' (see above) except that contour lines generated for the surface can be smoothed where they cross soft breaks.

A TINElement of type ControlContour behaves as a 'breakline' (see above). The z coordinate values are identical for all points in the 'control contour' LineString geometry. Triangles within the vicinity of a control contour need to be assessed during re-triangulation to ensure that they are not zero slope triangles (all three vertices fall on the same control contour).

Voids enclose a voided area of the TIN surface. They can be represented by either break, drape or (regular) void types of TINElement. Void geometry is of type Polygon, which might include interior boundaries. Triangles within a void still exist in the TIN surface but are considered to be void. Triangles inside of the Polygon interior boundary are in the exterior of the Polygon and therefore not in the void: these triangles are therefore not considered to be void.

For a void TINElement of type BreakVoid the boundary LineStrings of the 3D Polygon behave as breaklines in that triangles in the simple triangle patch collection must be adjusted so that no triangle is crossed by the break void boundary. Part or all of the break void boundary becomes an edge of two or more triangles. The elevation of this break void boundary takes precedence over the elevation of the original triangulated surface for the entire length of the boundary.

For a void TINElement of type DrapeVoid the boundary LineStrings of the 2D or 3D Polygon behave as breaklines in that triangles in the simple triangle patch collection must be adjusted so that no triangle is crossed by the drape void boundary. Part or all of the drape void boundary becomes an edge of two or more triangles. However, for drape voids, the elevation of the original triangulated surface takes precedence over the elevation of the drape void boundary.

For a void TINElement of type (regular) Void the boundary LineStrings of the 3D Polygon behave as breaklines in that triangles in the simple triangle patch collection must be adjusted so that no triangle is crossed by the void boundary. Part or all of the void boundary becomes an edge of two or more triangles. However, for regular voids, only the elevations of the void boundary vertices take precedence over the elevation of corresponding points on the original surface. That is, these vertices are treated as points for triangulating. The regular void boundaries between these vertices are handled as drape void boundaries - elevations from the original surface take precedence.

A hole is an area of a TIN surface, defined by a 2D or 3D Polygon, which is to be treated as a hole in the surface in that the triangles within this area still exist but are considered to

be part of the hole. The difference between a void and a hole is realized when two TIN surfaces are merged. When merging TIN surface A with TIN surface B, a void in surface A will take precedence over what is in the same area in B, the result being retention of the voided A triangles. A hole in surface A results in that part of surface B showing through the hole and becoming part of the resultant merged surface. Merging of TINs is relevant in LandInfra in the construction of SurfacesLayers (7.9.6).

Hole boundaries are different than an interior boundary of a surface. The hole is still part of the interior of the surface since the triangles still exist. They are merely considered as being in the hole so applications know how to deal with them in a visibility context. This also allows for "islands" to exist inside the holes (if the hole polygon has interior rings): here the triangles exist and are not hole triangles. Had the triangles in the hole been eliminated, islands could not be supported in a single surface.

A TINElement of type Hole encloses an area of the TIN surface designated as a hole. When a hole is specified for TINElement, the boundary LineStrings of the 2D or 3D Polygon behave as breaklines in that triangles in the simple triangle patch collection must be adjusted so that no triangle is crossed by the hole boundary. Part or all of the hole boundary becomes an edge of two or more triangles. Hole boundaries are treated like drape void boundaries in that the elevation of the original triangulated surface takes precedence over the elevation of the hole boundary, if present.

A TINElement of type StopLine is used to specify areas where the local continuity or regularity of the TIN surface is questionable. It is implementation-defined whether triangles in the simple triangle patch collection whose boundaries are (2D) crossed by a stop line are removed from the collection of simple triangle patches or are retained but enclosed within a DrapeVoid type of TINElement.

## 7.10 Requirements class: LandDivision

Requirements Class	
<b>/req/land-division</b>	
Target type	Encoding of the conceptual model
Name	Land Division
Dependency	<b>/req/land-infra</b>
Requirement	<b>/req/land-division/classes</b>  The LandDivision Requirements Class Classes shown in blue in Figure 60 shall be provided for by the encoding in a manner consistent with the encoding.

A LandInfra dataset may refer to boundaries delimiting ownership in land or in alternative terms: boundaries of real estate, of real property or of immovable property. LandXML applied the term “LandParcel” and this term is adopted for use in this standard. Land parcels are not the only division of the surface of the Earth, as the sovereign State which protects ownership is bordered and furthermore divides its territory according to political, judicial, or executive points of view. Finally, ownership in land is not confined to the surface of the Earth as buildings may be subdivided into condominiums, alternatively: strata title, through a condominium or strata title scheme, and jurisdictions may attribute ownership to subsurface structures like tunnels.

The above-mentioned domain is modelled through ISO 19152:2012 – Land Administration Domain Model (LADM). This ISO standard regards ‘the process of determining, recording and disseminating information about the relationship between people and land’ and offers a rich conceptual scheme which facilitates shared description of formal or informal practices concerning public recording of interests in land, including below the surface, and above the ground. The three packages of the standard regard parties (people and organizations); basic administrative units, rights, responsibilities, and restrictions (ownership rights); spatial units (parcels, and the legal space of buildings and utility networks) with a sub package for surveying and representation (geometry and topology).

This LandInfra standard addresses land development in the context of activities concerning civil engineering infrastructure facilities. The focus is therefore on determining and surveying of boundaries and the legal and administrative concepts needed to define the bordered entities. The scope of LandInfra thus does not include land recording and database storage, including interoperability of national recordings, and trans-national recording exchanges, but it does include interchange of survey plan data in terms of documents which are to be lodged at national recording agencies. These datasets are modelled as Statements with Signatory, e.g. owner or professional, and the signing

date. This subset of the LADM includes the LADM package spatial units, while the package basic administrative (read: real property) units and ownership and lesser rights is rendered through a single class. This class: InterestInLand, is documented by a Statement and refers to a SpatialUnit. The Statement compares to the abstract LADM class LA\_Source, which refers to documents with legal or surveying content. LandInfra does not distinguish between LA\_AdministrativeSource and LA\_SpatialSource, as e.g. a signed, cadastral plan is taken to have the same quality in legal and other respects as official letters. As regards to surveying and spatial representation, a number of LADM classes have been consolidated into the class BoundingElement, to further reduce the complexity of this LandInfra Standard.

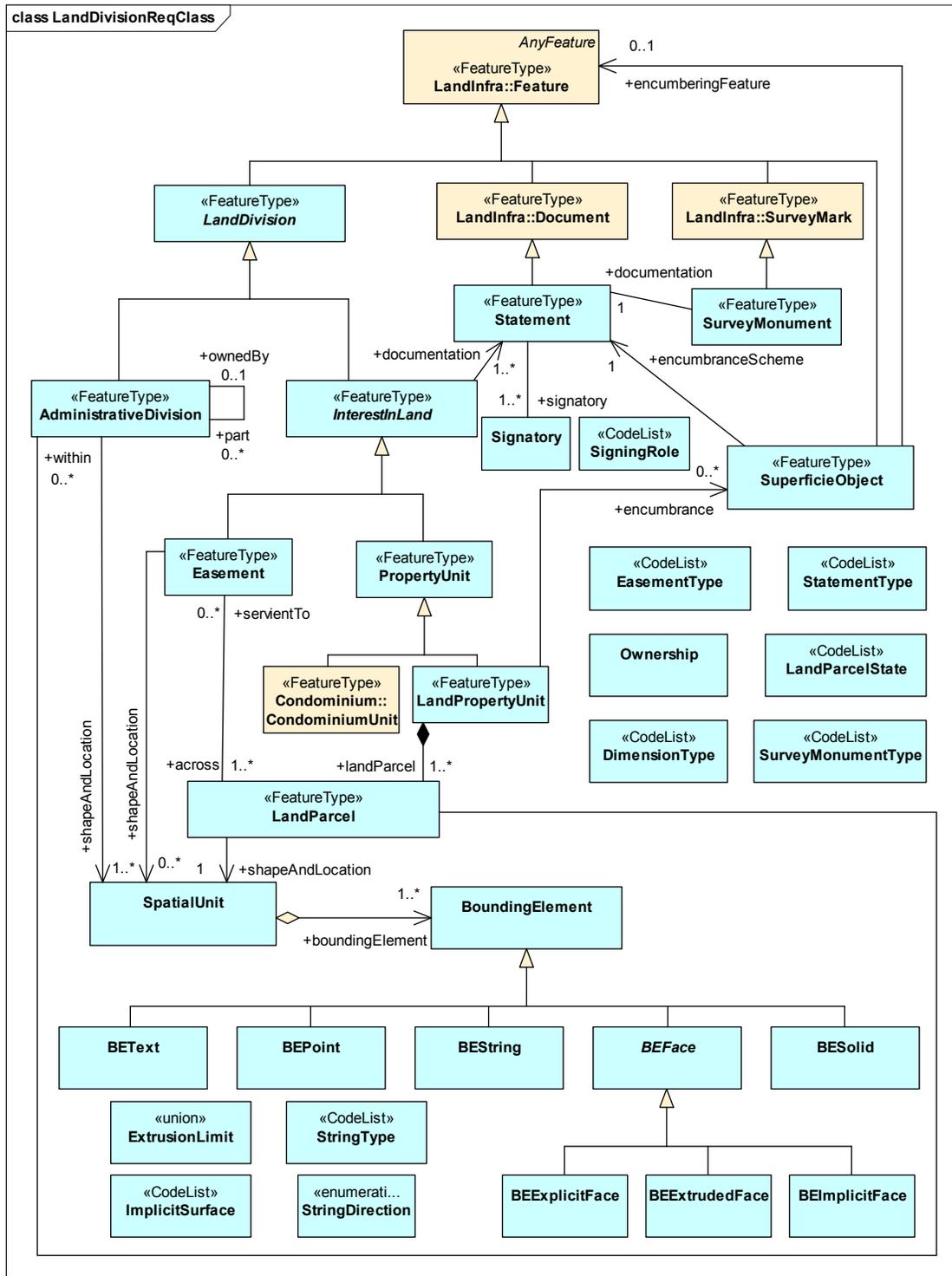


Figure 60. LandDivision Requirements Class

## 7.10.1 LandDivision

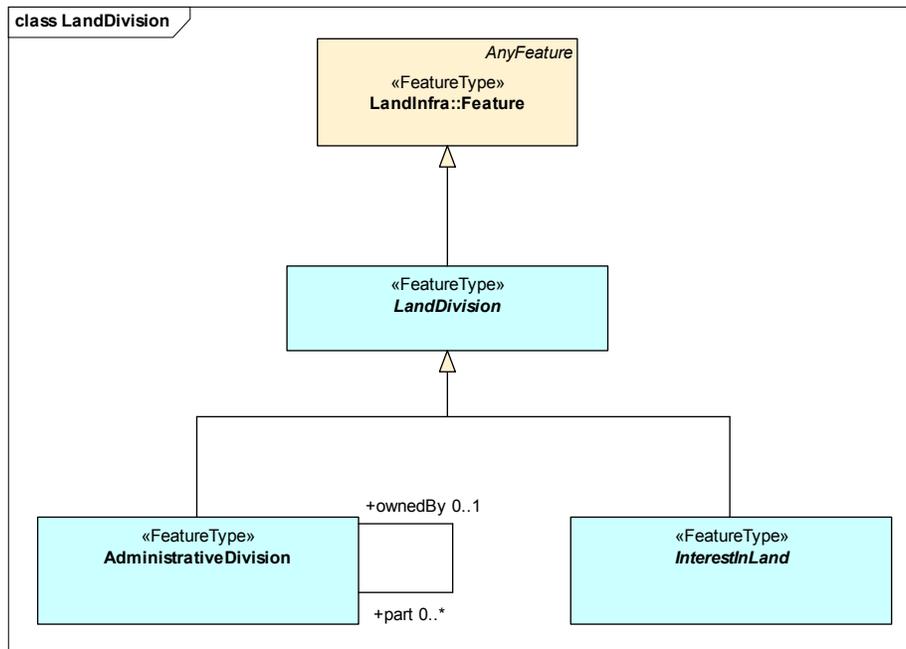


Figure 61. LandDivision context diagram

Land can be divided up into LandDivisions. These can either be public (political, judicial, or executive) or private in nature. The former are AdministrativeDivisions and the latter are InterestsInLand. The geometrical shape of these spatial units is modelled independently from their administrative or legal status, (see 7.10.6 SpatialUnit). As a subtype of Feature, LandDivision inherits attributes featureID, name, description, spatialRepresentation, linearlyReferencedLocation, property, and propertySet.



An InterestInLand is ownership or security towards real property [12]. Ownership of land extends to the ownership of all buildings stably erected on the land, as well as fixtures and plants. As a general rule with many exceptions, the vertical scope of ownership extends to the earth below the land, and to the sky above the land as is needed for the enjoyment of that land [20]. Ownership includes the right to give a lease, an easement, or a security interest. Easement stands out, because it generally does not apply to the same spatial borders as ownership. Therefore, the type Easement is introduced (see 7.10.2.7). This type may be applied also for profit à prendre (the right to take something off the land of another), for leases which do not apply to a whole PropertyUnit, and for certain charges by determination of law. The context diagram for this is shown in Figure 62.

The abstract class InterestInLand is documented by a Statement, while the shape and location of an InterestInLand is defined by SpatialUnit. InterestInLand types include PropertyUnits and Easements in the above-mentioned wide sense. The type and number of SpatialUnits depends upon the type of InterestInLand.

As a subtype of Feature, InterestInLand inherits attributes featureID, name, description, spatialRepresentation, linearlyReferencedLocation, property, and propertySet. An InterestInLand also has the following attribute:

**documentation:** one or more Statements

### 7.10.2.1 PropertyUnit

A PropertyUnit is a unit of ownership in land. PropertyUnit compares to the ‘basic administrative unit’, LA\_BAUnit, of the above mentioned LADM standard. In some jurisdictions, a land parcel is a single, contiguous area of land, possibly including water. In other jurisdictions, a single land parcel may contain several disjoint areas of land. To accommodate this difference, this LandInfra Standard considers each separate contiguous area to be a LandParcel, as automation requests a machine-readable identification of all lots. For the former case, the PropertyUnit contains a single LandParcel. For the latter case, the PropertyUnit is a collection of multiple LandParcels.

Ownership in land (as specified by a PropertyUnit) includes buildings and fixtures on the land parcel. However, lawful process may establish units of real property which are not bound to the surface of the Earth, namely in terms of CondominiumUnits through a CondominiumScheme (see 7.11), or in terms of SuperficieObjects through an encumbranceScheme (see 7.10.2.6).

In addition to the attributes inherited from InterestInLand, PropertyUnit has the following attributes:

**propertyUnitID:** identifier of a PropertyUnit, unique across an entire jurisdiction or globally unique with the inclusion of an ID.scope value

**ownership:** one or more Ownerships of the PropertyUnit

**postAddress:** zero or more post addresses of the PropertyUnit

### 7.10.2.2 Ownership

Ownership specifies the name of one or more owners of the PropertyUnit (and therefore all of its LandParcels or each of the CondominiumUnits or SuperficieObjects), as specified in a Statement concerning the PropertyUnit or recorded by the agency concerned. Ownership has the following attributes:

**name:** name of the Owner

**relativeShare:** percentage owned

**ownerAddress:** post address of the Owner

Note 1 to relativeShare: if there is a single owner, then the relative share is optional and assumed to be 100

Note 2 to ownerAddress: if the post address of the owner(s) is the post address of the PropertyUnit being owned, then the ownerAddress is optional

### 7.10.2.3 LandPropertyUnit

As a type of PropertyUnit, LandPropertyUnit specifies ownership to a part of the surface of the Earth. In addition to the attributes inherited from InterestInLand and PropertyUnit, LandPropertyUnit has the following attributes:

**landParcel:** one or more LandParcels which together constitute the LandPropertyUnit

**encumbrance:** zero or more SuperficieObjects which are recorded encumbrances against the LandPropertyUnit

### 7.10.2.4 LandParcel

A LandParcel is a part of the surface of the Earth. To specify ownership, LandParcel is a type of LandPropertyUnit. A LandParcel may have different states, among others to accommodate the situation where the LandParcel is an accessory part of condominium ownership. The shape and location of a LandParcel is specified by a single SpatialUnit.

As a subtype of Feature, LandParcel inherits attributes featureID, name, description, spatialRepresentation (though SpatialUnit would be a better choice for this),

linearlyReferencedLocation, property, and propertySet. A LandParcel has the following additional attributes:

**cadastralParcelID:** identifier of a LandParcel, unique across an entire jurisdiction (or globally unique with the inclusion of an ID.scope value), as specified in a Statement of type parcelEstablishment, where Signatory is or becomes owner of the PropertyUnit or as issued by the recording authority

**landRecordingId:** optional caseId, issued by the agency concerned

**parcelArea:** area of the LandParcel

**currentLandUse:** optional, jurisdiction-specific enumeration, e.g. of actual land use (cropland, forest, grassland, wetland, settlements, ...)

**plannedLandUse:** optional spatial planning land use categories (urban, rural, coastal zone, ...)

**parcelState:** optional LandParcelState

**within:** optional reference to AdministrativeDivision(s)

**servientTo:** zero or more Easements across the LandParcel, relevant e.g. where the footprint of a projected construction has to respect existing easements.

**shapeAndLocation:** the SpatialUnit that defines the single, contiguous geometry and location of the LandParcel

Note 1 to parcelArea: SurveyType, cf. 7.8.1.2, provides a means for recording data source

Note 2 to shapeAndLocation: String options of BoundingElements (3c and 3d), the above **within** reference and 7.9.2.1 LandElementType may allow for jurisdiction-specific edge-wise coding of parcel boundary

#### 7.10.2.5 Land Parcel State

LandParcelState has the following values:

**firstRegistration:** initial mapping of LandParcels or filling a void in existing recordings

**transitory:** LandParcel surveyed to be alienated from a current LandParcel, added to a neighboring LandParcel, and then extinguished, while its area is added to the said neighboring LandParcel

**current:** LandParcel as documented or as resurveyed

**carrierParcel:** current LandParcel on which a CondominiumBuilding is located or which is encumbered with a SuperficieObject

**mainParcel:** optional indication of a current LandParcel which is part of a LandPropertyUnit with more LandParcels, and which is chosen as reference for all LandParcels in the LandPropertyUnit. May be applied in jurisdictions where legal provisions do not recognize the concept of LandPropertyUnit

**extinguished:** LandParcel record not referring to current state of affairs

### 7.10.2.6 SuperficieObject

A SuperficieObject [21] comprises a building or other construction, including pipes, cables or tunnels, established and owned by a party (other than the owner of the LandParcels on which the superficie object is constructed) according to a granting Statement. The granting Statement may be issued by the owner of the LandParcels concerned, or according to statute. Ownership granting may take place before the object is established in situ. Subsurface SuperficieObjects may comprise a volume of soil surrounding the construction. As a subtype of Feature, SuperficieObject inherits attributes featureID, name, description, spatialRepresentation, linearlyReferencedLocation, property, and propertySet. The SuperficieObject has the following additional attributes:

**superficieID:** identifier of the SuperficieObject, unique across an entire jurisdiction, or globally unique with the inclusion of an ID.scope value

**type:** a jurisdiction-specific enumeration of different types of SuperficieObject

**ownership:** name of one or more owners of the SuperficieObject, as specified in an encumbrance Statement. In case of more than one owner, their relative share is recorded as well, along with the owners' addresses.

**encumbranceScheme:** the Statement, which establishes the right in the SuperficieObject

**encumberingFeature:** link to the encumbering feature, if it is already an instance of type LandInfra.Feature, e.g., a FacilityPart

A SuperficieObject Feature can be instantiated in accordance with the LandDivision Requirements Class and associated with a PropertyUnit also specified by that RC. If an implementation only supports the LandDivision RC (and the LandInfra core RC), this will suffice. But if the implementation also supports the Facility RC, the superficie object may already exist within the same LandInfra dataset containing the PropertyUnit,

but as a FacilityPart (subtype of LandInfra.Feature). The solution would be to still create an instance of SuperficieObject, but to then specify the FacilityPart as the SuperficieObject's encumberingFeature.

### 7.10.2.7 Easement

An Easement grants the beneficiaryParty right to a certain use, e.g. right of way, as specified by EasementType, within an area or volume delimited by one or more SpatialUnits, across one or more instances of what is called a servient LandParcel. This type may be used to account for rights and restrictions which are wider than the common law conception of easement, including e.g. leases concerning underground objects.

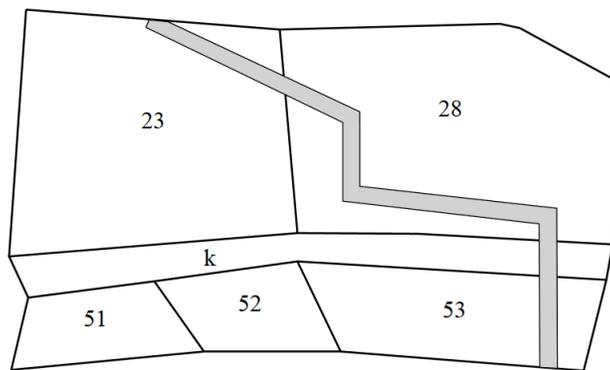


Figure 63. An Easement (gray) across four LandParcels

As a subtype of Feature, Easement inherits attributes featureID, name, description, spatialRepresentation (though SpatialUnit would be a better choice for this), linearlyReferencedLocation, property, and propertySet. An Easement has the following attributes:

- easementID:** identifier of the Easement, unique across an entire jurisdiction, or globally unique with the inclusion of an ID.scope value
- type:** a jurisdiction-specific enumeration of different types of easement
- beneficiaryParty:** optional identification of beneficiary parties
- permittedUse:** context-specific description of the permitted use
- across:** one or more servient LandParcels which the Easement encumbers
- shapeAndLocation:** zero or more SpatialUnits that define the geometry and location of the Easement

Note 1 to shapeAndLocation: of no value is provided, the Easement is assumed to be across the entire servient LandParcel(s) whose shapeAndLocation are already known

### 7.10.3 AdministrativeDivision

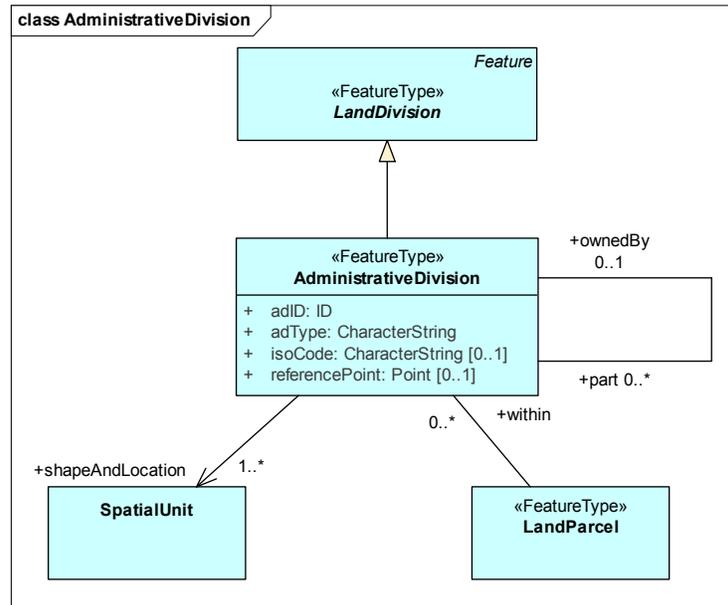


Figure 64. AdministrativeDivision Context Diagram

State territory is divided according to political, judicial, or executive points of view. ISO 3166-2 defines codes for the names of the principal (political) subdivisions (e.g., provinces or states) of all countries coded in ISO 3166-1. Normally, the division is hierarchical, e.g. in terms of a court hierarchy, or regional and local units of the central executive power, e.g. in terms of police or health and safety bodies. AdministrativeDivision is also a means of verbal localization of LandParcels.

As a subtype of Feature, AdministrativeDivision inherits attributes `featureID`, `name`, `description`, `spatialRepresentation` (though `SpatialUnit` would be a better choice for this), `linearlyReferencedLocation`, `property`, and `propertySet`. AdministrativeDivision has the following additional attributes:

**adID:** identifier of the AdministrativeDivision

**adType:** type of the AdministrativeDivision, e.g. municipality or local court

**isoCode:** optional ISO 3166-2 subdivision code

**adName:** optional geographical name of the Administrative Division

**shapeAndLocation:** one or more SpatialUnits that define the geometry and location of the AdministrativeDivision

**part:** if the AdministrativeDivision is subdivided (e.g., Country into States), the parts resulting from the division

**ownedBy:** if the AdministrativeDivision is the result of the division of a larger AdministrativeDivision, then the owning AdministrativeDivision which was divided

Note 1 to entry: A representative point of the AdministrativeDivision is available through shapeAndLocation (BoundaryElement +point)

Note 2 to entry: Enclaves of an instance of AdministrativeDivision within another instance (with hole) may be rendered through more BoundaryElements and BoundaryElement +ring of SpatialUnit, respectively.

#### 7.10.4 Statement

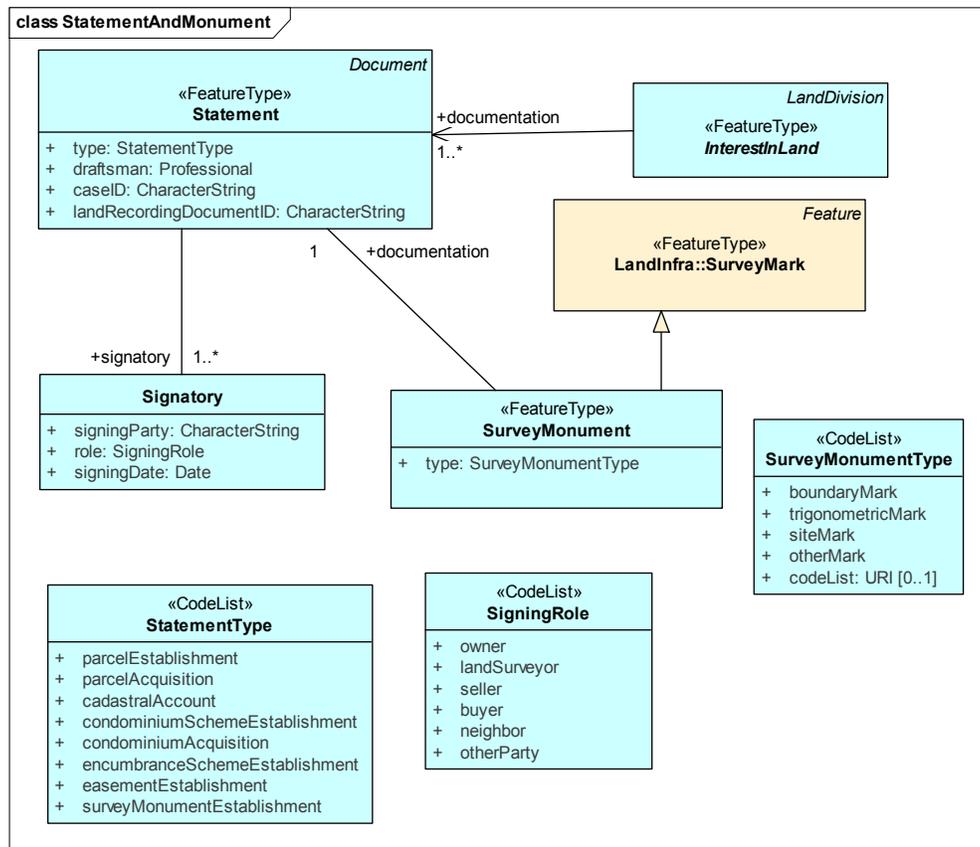


Figure 65. Statement and SurveyMonument context diagram

A Statement documents the establishment or acquisition of an InterestInLand or a SurveyMonument in accordance with the specific StatementType. A Statement document is signed by one or more Signatories, each with a specific SigningRole. The context diagram for Statement appears as Figure 65.

As a subtype of Feature, Statement inherits attributes featureID, name, description, spatialRepresentation, linearlyReferencedLocation, property, and propertySet. As a subtype of Document, Statement inherits the attributes of documentID and documentType. Statement has the following additional attributes:

**type:** the type of Statement

**draftsman:** the professional (lawyer, notary, landSurveyor, ...) who prepared (drafted) the Statement

**caseID:** the draftsman's case identification, usually quoted on the bill and in communication with authorities

**landRecordingDocumentID:** the document identification issued by the concerned land recording agency (land registry, cadaster, or land administration agency, as dependent on jurisdiction), as the Statement is assumed to have been or to become recorded

**signatory:** information about the one or more signatures on the Statement

#### 7.10.4.1 StatementType and SigningRole

InterestsInLand are established, acquired, changed and abolished. The requested StatementTypes are limited to establishment and acquisition. Abolishment of an interest is considered to be out of scope for this standard.

SigningRole specifies the role that a signing party plays in the signing of the Statement.

The supported StatementTypes and their associated SigningRoles are as follows:

**parcelEstablishment:** the initial cadastral survey, or the creation of a new LandParcel through change of existing LandParcels. Signatory roles: LandParcel owner (optional) and landSurveyor

**parcelAcquisition:** transfer of ownership to a LandParcel from the seller to the buyer

**cadastralAccount:** landSurveyor's account of parcel and boundary changes, with references to pertinent legislation, intended for recording body (land administration agency, cadastral authority). Signatory role: landSurveyor

**condominiumSchemeEstablishment:** subdivision of a multi-storage building into ownership units through a scheme, which for each unit specifies one or more BuildingSpatialUnits and a share in the joint facilities. Signatory roles: PropertyUnit owner and landSurveyor

**condominiumAcquisition:** transfer of ownership from a seller to a buyer, where seller is either the owner of the original PropertyUnit or the owner of a CondominiumUnit

**encumbranceSchemeEstablishment:** granting of ownership rights by the owner of a LandPropertyUnit, or according to statute, to an existing or intended SuperficieObject through a scheme which specifies one or more LandParcels. Signatory roles: LandPropertyUnit owner (if applicable) and landSurveyor

**easementEstablishment:** the owner of a servient LandParcel grants another party a certain right, as specified by easementType and permittedUse, and geographically delimited by SpatialUnit. The other party may be one or more current owner(s) of dominant LandParcels in the vicinity of the servient LandParcel, or e.g. facility owners like a power company or a water company. The owner association according to a CondominiumScheme may grant to a CondominiumUnit an Easement, e.g. in terms of a parking lot on the LandParcel around the CondominiumBuilding. Signatory roles: Owner of the servient LandParcel or the mentioned owner association, and optionally the other party and landSurveyor.

**surveyMonumentEstablishment** is signed by the owner of the concerned instance of LandParcel, and optionally by the neighbor (the owner of the neighboring LandParcel) and the landSurveyor.

### 7.10.5 SurveyMonument

A SurveyMonument is a physical object, the location of which is stable and by its form defines a point on the surface of the Earth. The SurveyMonument may carry a symbol of authority, which communicates the purpose of the monument. SurveyMonument is a specialization of SurveyMark. The context diagram for SurveyMonument is in Figure 65.

As a subtype of Feature, SurveyMonument inherits attributes featureID, name, description, spatialRepresentation, linearlyReferencedLocation, property, and propertySet. As a subtype of SurveyMark, SurveyMonument inherits the attributes of identification and the Feature.SpatialRepresentation is limited to Point. Additional attributes of SurveyMonument are:

**type:** the SurveyMonumentType

**documentation:** the Statement regarding the establishment of the SurveyMonument

### 7.10.5.1 SurveyMonumentType

SurveyMonumentType specifies the allowable values for type of SurveyMonument. These are:

**boundaryMark:** marking of the legal boundary between two LandParcels according to jurisdiction-specific provisions

**trigonometricMark:** marks an observation point of geodetic significance.

**siteMark:** marks an observation point in the context of a building site, either as a verticalSiteMark or a horizontalSiteMark, or both

**otherMark:** e.g. for marking of easement boundaries or for temporary marking of legal boundary points with wood pegs or similar.

The inclusion of the optional tagged value codeList: URI choice, is in accordance with ISO 19103. If present, then the only valid values are those from an external authority referenced code list identified by the URI value. If not, the other CodeListed values are valid along with possible additional, user defined values. However, these additional values should not replace an existing code by changing the name or definition, or have the same definition as an existing value.

### 7.10.6 SpatialUnit

SpatialUnit is a contiguous geometrical entity, which is delimited and located on or close to the surface of the Earth through its BoundingElements. The dimension of SpatialUnit is specified by its DimensionType. The context diagram for SpatialUnit is in Figure 66.

SpatialUnit has the following attributes:

**spatialUnitID:** user defined ID, unique within a jurisdiction, or globally unique with the inclusion of an ID.scope value

**name:** jurisdiction-specific name for the SpatialUnit

**description:** optional description

**dimension:** 2D or 3D or liminal. Depending on DimensionType, SpatialUnit attributes may include area, volume and height.

**area:** optional area of the SpatialUnit

**volume:** optional volume of the Spatial unit if it is 3D

**height:** optional height of the Spatial unit if it is 3D

**boundingElement:** one or more BoundingElements which describe or define the geometry and location of the SpatialUnit

Liminal dimension applies when a spatial unit is bounded by both 2D and 3D spatial units.

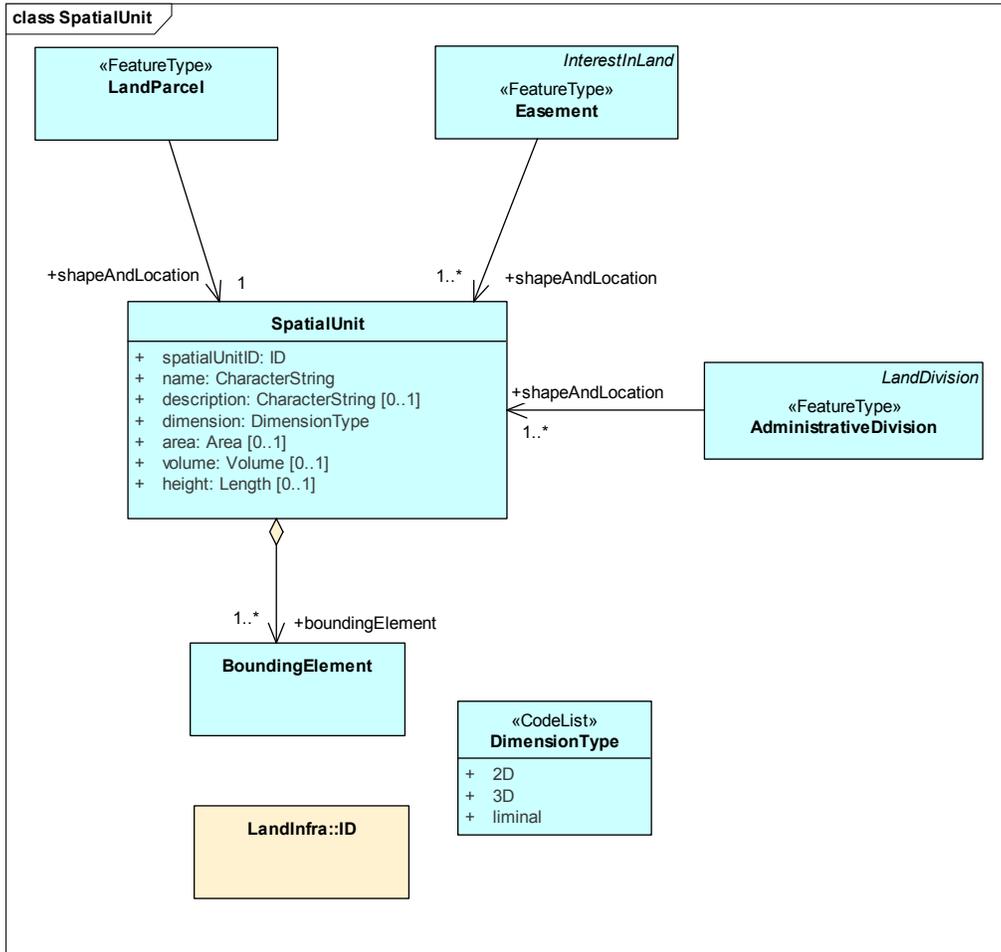


Figure 66. SpatialUnit context diagram

## 7.10.7 BoundingElements

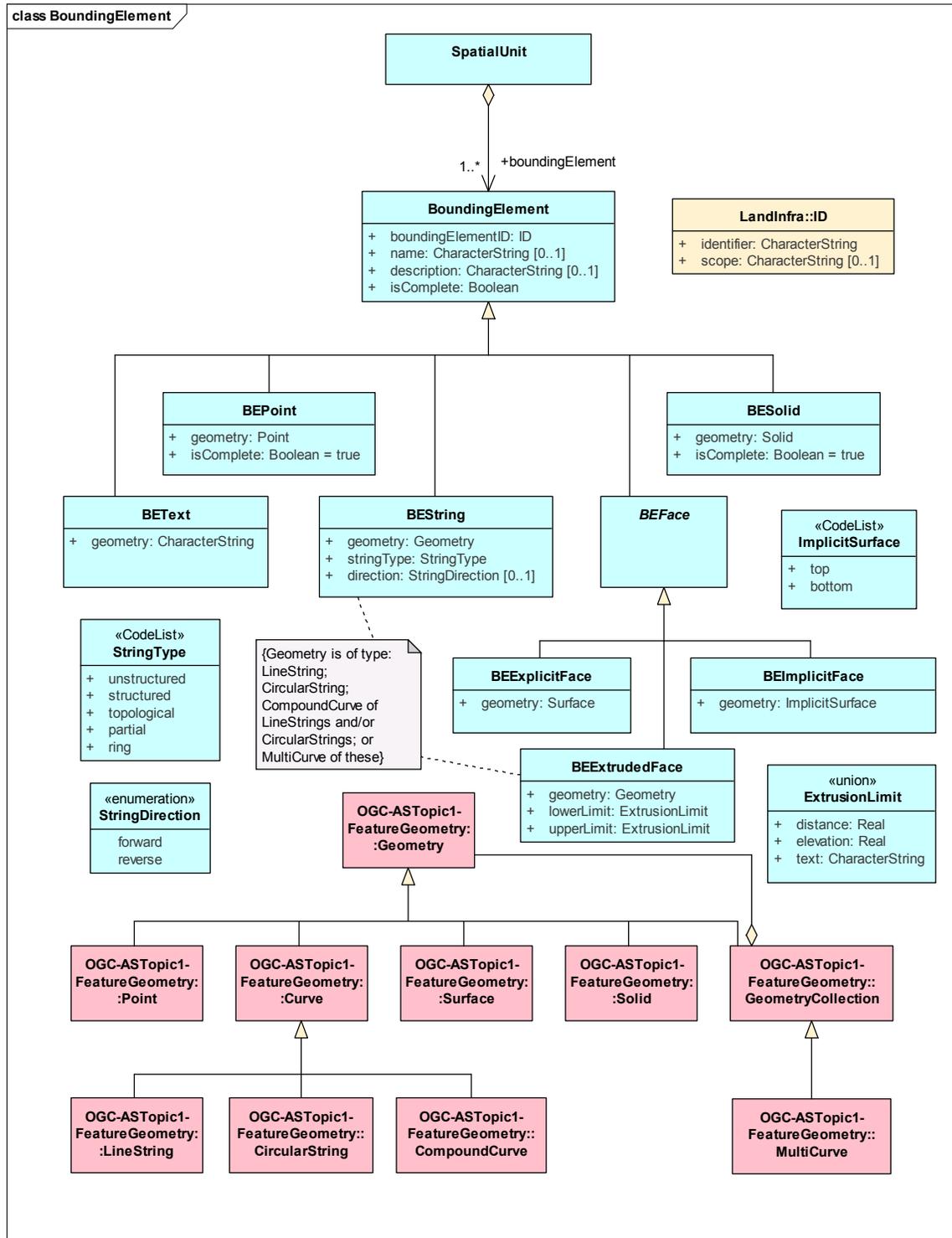


Figure 67. BoundingElement context diagram

The LandParcel Requirements Class Classes simplify the legal-administrative variety of LADM basic administrative units, rights, responsibilities, and restrictions (ownership rights). Similarly, LADM spatial units (parcels and the legal space of buildings and utility networks) are simplified through the notion of BoundingElement. The context diagram for BoundingElements is shown in Figure 67.

A BoundingElement provides all or part of the boundary of a SpatialUnit. If it provides the entire boundary, then the isComplete Boolean attribute value is TRUE. BoundingElement can take any of the following forms (derived from LADM). The value of isComplete can be TRUE or FALSE unless otherwise noted below.

- 1) **text** - the BoundingElement is described with text instead of having a geometric representation, (e.g., a legal “meets and bounds” description or a less formal description such as “from the big rock to the old oak tree”) where the text can describe either string or face boundaries
- 2) **point** - the BoundingElement of the SpatialUnit has a zero-dimensional geometric representation (a single representative Point)
- 3) **string** - the BoundingElement has a one-dimensional geometric string representation, limited to curves of type LineString; CircularString; CompoundCurve constructed from LineStrings, CircularStrings, and/or other CompoundCurves; or MultiCurve of any of the previously specified types. The normal convention is followed for direction of geometric boundary element geometries: for exterior boundary elements, the direction shall be counterclockwise (anti-clockwise) and for interior boundary elements (holes), the direction shall be clockwise. The optional string direction attribute allows the curve to be geometrically defined in one direction but then used as a BoundingElement in the reverse direction.
  - 3a) **unstructured curve** - the BoundingElement linework may not be internally contiguous or contiguous with other BoundingElements of the same **SpatialUnit** due to under- and overshoots (isComplete = FALSE)
  - 3b) **structured curve** - the BoundingElement linework is internally contiguous and expected to be contiguous with other BoundingElements of the same SpatialUnit (which would therefore also have to be structured curves)
  - 3c) **topological curve** - the BoundingElement is the common SpatialUnit edge shared by adjacent SpatialUnits; linework is internally contiguous and expected to be contiguous with other BoundingElements of the same and adjacent SpatialUnits (which would therefore also have to be contiguous curve based) (isComplete = FALSE). Once specified as a BoundingElement for some SpatialUnit, it can be re-used elsewhere by merely providing its id and possibly changing its direction.

- 3d) **partial curve** - the BoundingElement only uses part of the provided geometric representation (e.g., a single geometric representation of a coastline may be used as a BoundingElement for all of the SpatialUnits along the coastline) (isComplete = FALSE). Once specified as a BoundingElement for some SpatialUnit, it can be re-used elsewhere by merely providing its id and possibly changing its direction.
- 3e) **ring** - the curves of the sole BoundingElement of the SpatialUnit form a ring or set of rings (simple and closed geometries) sufficient to define the entire boundary of a polygon or polyhedral surface (isComplete = TRUE). The LineString, CircularString, CompoundCurve, or first curve in a MultiCurve must be rings, and are interpreted to be the exterior bounding ring of the SpatialUnit. For holes, only a MultiCurve containing rings is allowed, and all but the first ring in the MultiCurve shall be interpreted as interior boundary rings.
- 4) **face** - the BoundingElement has a two-dimensional, vertical or nearly vertical (except for implicit surfaces which are horizontal or nearly horizontal) geometric surface representation (extruded, explicit, or implicit polygon or polyhedral surface)
- 4a) **extruded surface** - the BoundingElement is defined by a string as above, but which is extruded between a lower and upper limit, each of which may be defined by a fixed vertical distance up (positive) or down (negative) from the string, by a constant elevation value, or by a text string such as “unlimited”. The upper and lower limits cannot be the same. If either is equivalent to the string itself, it should be specified with a distance = 0.0.
- 4b) **explicit surface** - the BoundingElement is defined by a polygon or polyhedral surface
- 4c) **implicit surface** - the BoundingElement represents the top or bottom of a SpatialUnit which is not explicitly defined but instead derived from the BoundingElements which represent all of the sides of the SpatialUnit
- 4d) **extruded topological surface** - the BoundingElement is the extrusion-defined common SpatialUnit side shared by adjacent SpatialUnits
- 4e) **explicit topological surface** - the BoundingElement is the explicitly defined common SpatialUnit side shared by adjacent SpatialUnits
- 4f) **implicit topological surface** - the BoundingElement is the implicitly defined common SpatialUnit top or bottom shared by vertically adjacent SpatialUnits
- 5) **solid** - the sole BoundingElement of the SpatialUnit has a three-dimensional geometric representation (one or more simple, closed surfaces called shells) (isComplete = TRUE)

The set of BoundingElements that make up a SpatialUnit shall be consistent with the SpatialUnit dimension: 2D, 3D, or liminal. For example, for a SpatialUnit to be 2D, all of its BoundingElements must be of type **point** or any of the **curve** subtypes. For a SpatialUnit to be 3D, all of its BoundingElements must be of type **face** or **solid**. A liminal SpatialUnit will have some combination of these types. A SpatialUnit having **text** type BoundingElement(s) may be 2D, 3D, or liminal.

## 7.11 Requirements class: Condominium

Requirements Class	
<b>/req/condominium</b>	
Target type	Encoding of the conceptual model
Name	Condominium
Dependency	<b>/req/land-division</b>
Requirement	<b>/req/condominium/classes</b>  The Condominium Requirements Class Classes shown in blue in Figure 68 shall be provided for by the encoding in a manner consistent with the encoding.

A Condominium is concurrent ownership of real property that has been divided into private and common portions. Condominium unit owners must be members of a mandatory owners' association and engage in the maintenance of joint facilities according to a specified share.

LandXML includes Building as a 'parcel format' and provides among other capabilities the recording of lot entitlements and liability apportionment for owners' corporation, body corporate or scheme land entity. ISO 19152:2012 – Land Administration Domain Model (LADM) similarly reflects a need for recording of condominium schemes through its LA\_LegalSpaceBuildingUnit. The OGC CityGML Standard allows for recording buildings and an Application Domain Extensions (ADE), the CityGML Immovable Property Taxation ADE [13], specifies relations between land parcel, building, and condominium parts. Finally, the IFC specifies building and their parts in detail. – The demonstrated need and the developed Immovable Property Taxation ADE motivated the inclusion into this standard of the Condominium Requirements class.

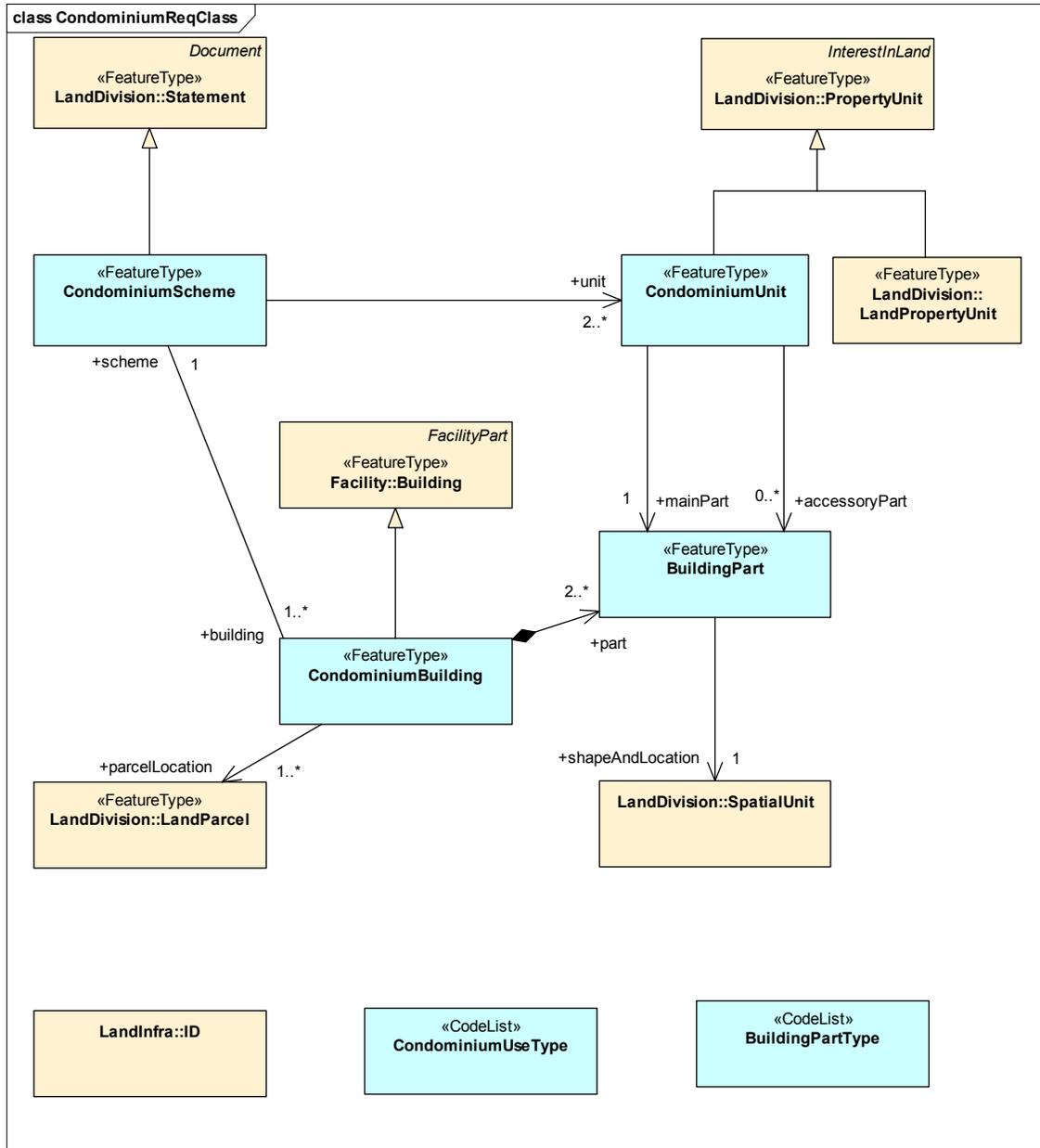


Figure 68. Condominium Requirements Class Classes

### 7.11.1 CondominiumUnit

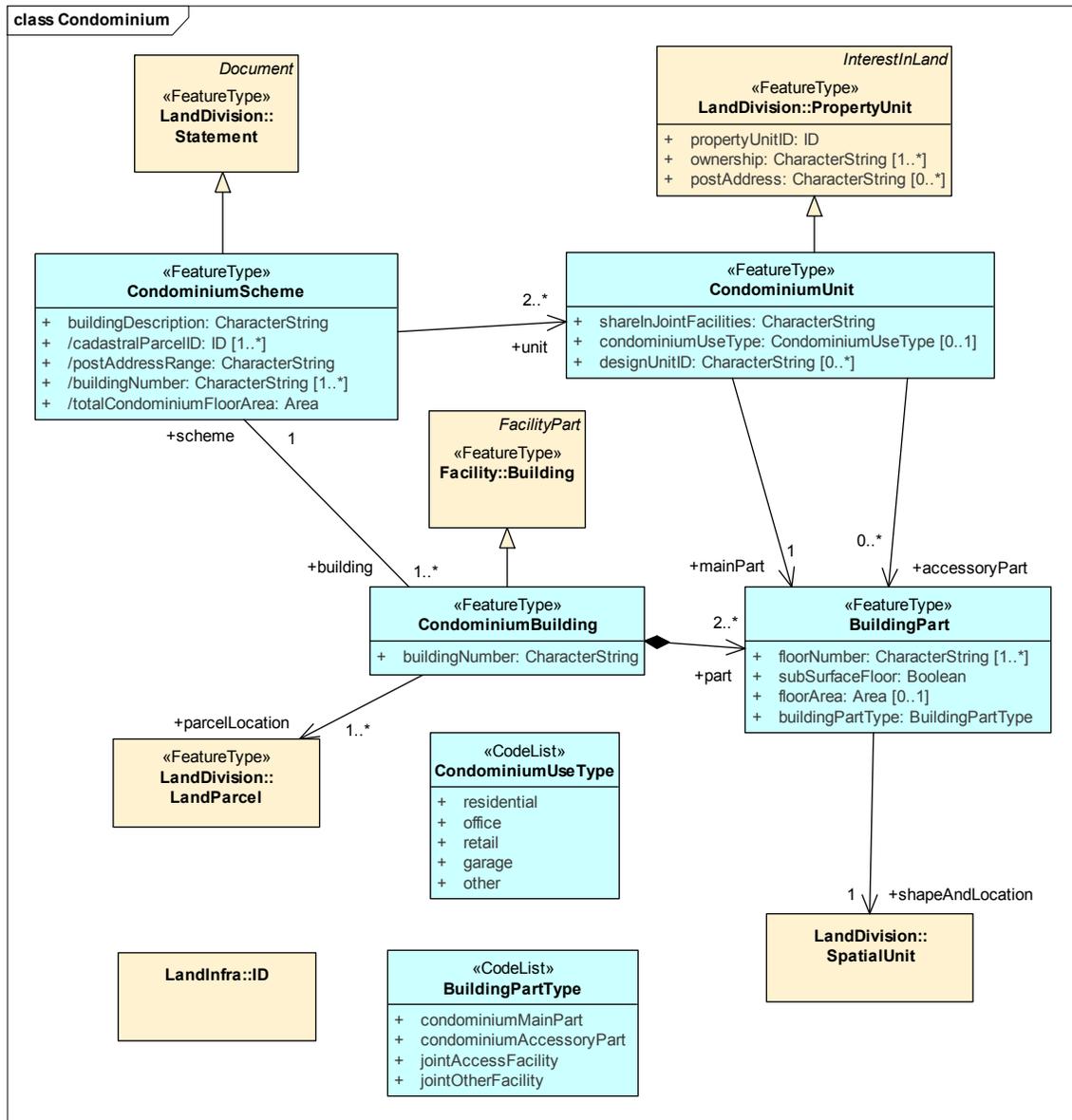


Figure 69. Condominium context diagram

A CondominiumUnit is a concurrent ownership of real property that has been divided into private and common portions [12, 14]. The privately owned part is made up of clearly demarcated parts of a building (BuildingPart), as specified by a CondominiumScheme. CondominiumUnit owners are members of a mandatory owners’ association. Moreover, each CondominiumUnit has a share in the common parts of the property which are also clearly demarcated. The context diagram for this is shown in Figure 69.

A CondominiumUnit is a specialization of PropertyUnit. The attributes of a CondominiumUnit are specified by the associated CondominiumScheme. In addition to attributes inherited from PropertyUnit (propertyUnitID, ownership, and postAddress), a CondominiumUnit has the following attributes:

**shareInJointFacilities:** the fraction by which the CondominiumUnit is bound to engage in the maintenance of joint facilities, including roof, outer walls, access areas, technical installations, and the LandParcel on which the building rests.

**condominiumUseType:** optional, jurisdiction-specific enumeration, e.g. of permitted use (habitation, office, shop, ...)

**designUnitID:** optional reference to the identification(s) used during the project phase

**mainPart:** the privately owned BuildingPart

**accessoryPart:** optional, additional, privately owned BuildingParts, for example, a garage in the basement or a shop on the ground floor.

### 7.11.2 CondominiumUseType

CondominiumUseType has the following optional values, reflecting jurisdiction-specific planning provisions:

**residential**

**office**

**retail**

**garage**

**other**

### 7.11.3 CondominiumBuilding

A CondominiumBuilding is a Building containing CondominiumUnits and therefore subject to a single CondominiumScheme which may comprise more buildings. Each CondominiumBuilding is located on one or, if legislation so permits, more LandParcels. The attributes of CondominiumBuilding are:

**buildingNumber:** identification for the CondominiumBuilding

**scheme:** the CondominiumScheme for the building(s)

**parcelLocation:** the LandParcel(s) on which the building is located

**part:** contiguous parts of a floor of the building, subdivided into exclusively owned parts and joint facility parts. The CondominiumScheme allocates exclusively owned main and optional exclusively owned accessory parts to each CondominiumUnit.

#### 7.11.4 BuildingPart

The CondominiumScheme subdivides the floors of a multi-storage CondominiumBuilding into BuildingParts according to management and use, thereby drawing upon the IfcSpatialZone class (with reference 5.4.3.51) of building SMART's Industry Foundation Classes. The BuildingParts intended for exclusive ownership have either type condominiumMainPart or condominiumAccessoryPart. The joint facilities are either of type jointAccessFacility e.g. staircase, lift or jointOtherFacility, e.g. laundry, heating facility. Each of the exclusively owned units has access to building entrance through one or more jointAccessFacilities. BuildingPart has the following attributes:

**floorNumber:** the CondominiumBuilding floor(s) on which the BuildingPart is located

**subsurfaceFloor:** True if the floor is below grade, otherwise False

**floorArea:** the optional floor area of the BuildingPart

**shapeAndLocation:** a single SpatialUnit which defines the geometry and location of the BuildingPart

Note to **subsurfaceFloor:** Working environment regulations may motivate more elaborate attributes, e.g. accounting also for rooms without natural lightning.

#### 7.11.5 BuildingPartType

BuildingPartType has the following values:

**condominiumMainPart:** the mandatory, privately owned BuildingPart to which the postAddress refers

**condominiumAccessoryPart:** A CondominiumUnit may in addition to the main part comprise additional privately owned BuildingParts, for example, a shop on the ground floor or a garage in the basement or in another building.

**jointAccessFacility:** The parts of the common portions used for access. Identification of these parts may be used for indoor navigation or security management.

**jointOtherFacility:** Remaining parts, including rooms containing technical facilities (heating, ventilation, etc.)

Note to **condominiumAccessoryPart:** Some jurisdictions allow the owner of a CondominiumUnit exclusive rights to a parking lot with surveyed boundaries on the concerned LandParcel. In this standard, such parking lot does not qualify as a LandParcel, because the condominium owner cannot alienate rights in the parking lot independently of the rest of the CondominiumUnit.

### 7.11.6 CondominiumScheme

A CondominiumScheme is a Statement which specifies the BuildingPart subdivision of multi-storage CondominiumBuildings into CondominiumUnits, and records the LandParcels on which the CondominiumBuildings rest. CondominiumScheme is specified in Figure 69.

As a type of Statement, CondominiumScheme inherits all Statement attributes. A CondominiumScheme has the following additional attributes:

**buildingDescription:** free-format text description of the floors and other components of the CondominiumBuilding(s), including its joint facilities

**unit:** two or more CondominiumUnits units included in the CondominiumScheme

**building:** one or more CondominiumBuildings subject to the scheme

From the list of CondominiumBuildings included in the CondominiumScheme, the following additional attributes can be derived for the CondominiumScheme:

**cadastralParcelId:** identification of the LandParcel(s) on which the CondominiumBuilding(s) containing the CondominiumUnits rests

**postAddressRange:** the range of postal addresses of the CondominiumBuildings containing the CondominiumUnits

**buildingNumber:** identification of the CondominiumBuilding(s) containing the CondominiumUnits

**totalCondominiumFloorArea:** sum of the Areas of the BuildingParts which define the individual CondominiumUnits

## Annex A Conformance Class Abstract Test Suite (Normative)

### A.1 Core Conformance class: LandInfra

<b>/conf/land-infra</b>		
Requirements	<a href="#">/req/land-infra</a>	
Dependency		
Test	<b>/conf/land-infra/dataset</b>	
	Requirement	<a href="#">/req/land-infra/dataset</a>
	Test purpose	Verify that the encoding specifies a LandInfra dataset in a format appropriate for that encoding.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b>/conf/land-infra/subjects</b>	
	Requirement	<a href="#">/req/land-infra/subjects</a>
	Test purpose	Verify that the encoding specifies which of the LandInfra subjects it supports.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b>/conf/land-infra/classes</b>	
	Requirement	<a href="#">/req/land-infra/classes</a>
	Test purpose	Verify that the encoding provides the LandInfra Requirements Class Classes shown in blue in Figure 2 in a manner consistent with the encoding.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b>/conf/land-infra/crs</b>	
	Requirement	<a href="#">/req/land-infra/crs</a>
	Test purpose	Verify that the encoding requirement for coordinate reference systems is consistent with OGC Abstract Specification Topic 2.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b>/conf/land-infra/featureID</b>	
	Requirement	<a href="#">/req/land-infra/featureID</a>
	Test purpose	Verify that the encoding specifies whether it supports ID at the feature level, ID at the Feature sub-type level, or both.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b>/conf/land-infra/19103</b>	
	Requirement	<a href="#">/req/land-infra/19103</a>

	Test purpose	Verify that the encoding provides the core data types specified in ISO 19103 Clause 7 and shown in pink in Figure 5 that are appropriate to the supported subject area(s).
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b><a href="#">/conf/land-infra/topic-1</a></b>	
	Requirement	<b><a href="#">/req/land-infra/topic-1</a></b>
	Test purpose	Verify that the encoding provides the geometry types specified in the OGC Abstract Specification Topic 1, Feature Geometry as shown in pink and the SimpleTriangle, TIN, and PolyfaceMesh extensions shown in beige in Figure 6, that are appropriate to the supported subject area(s).  Verify that the encoding provides the additional geometry types not found in Topic 1, but required by a specific supported requirements class, are specified in that requirements class.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b><a href="#">/conf/land-infra/topic-19-core</a></b>	
	Requirement	<b><a href="#">/req/land-infra/topic-19-core</a></b>
	Test purpose	Verify that, if the encoding supports linearly referenced locations, then the encoding provides the PositionExpression and related Classes specified in the OGC Abstract Specification Topic 19, core Linear Referencing (LR) Package shown in pink in Figure 7 that are appropriate to the supported subject area(s).
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b><a href="#">/conf/land-infra/topic-19-extensions</a></b>	
	Requirement	<b><a href="#">/req/land-infra/topic-19-extensions</a></b>
	Test purpose	Verify that, if the encoding supports linearly referenced locations, then the encoding specifies which of the OGC Abstract Specification Topic 19, Linear Referencing extension packages it supports and therefore supports the appropriate Classes shown in Figure 7 and Figure 8 for those packages that are appropriate to the supported subject area(s).
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

Test	<b><a href="#">/conf/land-infra/19109</a></b>	
	Requirement	<b><a href="#">/req/land-infra/19109</a></b>
	Test purpose	<p>Verify that Features shall be supported by a Land and Infrastructure encoding in accordance with the uml:feature requirement in ISO 19109:2015 Clause 8.2.6 and shown in Figure 9, as follows:</p> <p>Each instance of FeatureType shall be implemented by the encoding's equivalent of a UML Class having a generalization association with AnyFeature and with a stereotype of &lt;&lt;FeatureType&gt;&gt;.</p>
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

## A.2 Conformance class: Facility

<b>/conf/facility</b>		
Requirements	<a href="#">/req/facility</a>	
Dependency	<a href="#">/conf/land-infra</a>	
Test	<b><a href="#">/conf/facility/classes</a></b>	
	Requirement	<a href="#">/req/facility/classes</a>
	Test purpose	Verify that the encoding provides the Facility Requirements class Classes shown in blue in Figure 10 in a manner consistent with the encoding.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b><a href="#">/conf/facility/subtypes</a></b>	
	Requirement	<a href="#">/req/facility/subtypes</a>
	Test purpose	Verify that the encoding specifies which of the FacilityPart subtypes shown in Figure 10 it supports.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

### A.3 Conformance class: Project

<b>/conf/project</b>		
Requirements	<b>/req/project</b>	
Dependency	<b>/conf/facility</b>	
Test	<b>/conf/project/classes</b>	
	Requirement	<b>/req/project/classes</b>
	Test purpose	Verify that the encoding provides the Project Requirements Class Classes shown in blue in Figure 13 in a manner consistent with the encoding.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

#### A.4 Conformance class: Alignment

<b>/conf/alignment</b>		
Requirements	<a href="#">/req/alignment</a>	
Dependency	<a href="#">/conf/facility</a>	
Test	<b>/conf/alignment/classes</b>	
	Requirement	<a href="#">/req/alignment/classes</a>
	Test purpose	If an encoding includes Alignments, then verify that the encoding provides the Alignment Requirements Class Classes shown in blue in Figure 15 in a manner consistent with the encoding.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b>/conf/alignment/description</b>	
	Requirement	<a href="#">/req/alignment/description</a>
	Test purpose	Verify that an encoding which includes Alignments shall specify that all of the included Alignments shall be continuous, non-overlapping, and non-branching (though it may contain intersections with other Alignments) and that if it is used within the context of a Project, the included Alignment shall be for a single alternative, as specified by the ProjectPart.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b>/conf/alignment/CRS</b>	
	Requirement	<a href="#">/req/alignment/CRS</a>
	Test purpose	Verify that an encoding requires that all geometry/locations for Alignment2DHorSegments and Alignment2DVertSegments be specified using the Alignment2DHorizontal.crs which shall be a Cartesian local engineering reference system.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b>/conf/alignment/LR-interfaces</b>	
	Requirement	<a href="#">/req/alignment/LR-interfaces</a>
	Test purpose	Verify that the encoding implements the OGC Abstract Specification Topic 19 ILinearElement and ISpatial interfaces in a manner consistent with the encoding.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b>/conf/alignment/measures</b>	
	Requirement	<a href="#">/req/alignment/measures</a>

	Test purpose	<p>If an encoding uses Alignment as a linear element, verify that it specifies that:</p> <p>1) If an Alignment2DHorizontal.2DLineStringRepresentation or an Alignment3D.3DLineStringRepresentation is specified as a linear element, then distanceAlong shall be measured along the LineString and offsetLateralDistance and offsetVerticalDistance values shall be measured normal to the LineString from this DistanceAlong point.</p> <p>2) If an Alignment2DHorizontal.segment list of Alignment2DHorSegments is used as a linear element, then distanceAlong and offsetLateralDistance values shall be measured in the horizontal plane, ignoring any vertical displacement of the Alignment. OffsetVerticalDistance values shall be absolute from the horizontal plane unless an Alignment2DVertical is specified as a VerticalOffsetReferent so as to take into consideration vertical displacement of the Alignment, if present.</p>
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

## A.5 Conformance class: Road

<b>/conf/road</b>		
Requirements	<b>/req/road</b>	
Dependency	<b>/conf/facility</b>	
Dependency	<b>/conf/land-feature</b>	
Test	<b>/conf/road/classes</b>	
	Requirement	<b>/req/road/classes</b>
	Test purpose	Verify that the encoding provides the Road Requirements Class Classes shown in blue in Figure 21 in a manner consistent with the encoding.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b>/conf/road/alignment</b>	
	Requirement	<b>/req/road/alignment</b>
	Test purpose	Verify that if the encoding allows the linear element used for locating RoadElements to be an Alignment, then that encoding supports the Alignment Requirements Class.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

## A.6 Conformance class: RoadCrossSection

<b>/conf/road-cross-section</b>		
Requirements	<a href="#">/req/road-cross-section</a>	
Dependency	<a href="#">/conf/road</a>	
Test	<b>/conf/road-cross-section/classes</b>	
	Requirement	<a href="#">/req/road-cross-section/classes</a>
	Test purpose	Verify that the encoding provides the RoadCrossSection Requirements Class Classes shown in blue in Figure 25 in a manner consistent with the encoding.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b>/conf/road-cross-section/alignment</b>	
	Requirement	<a href="#">/req/road-cross-section/alignment</a>
	Test purpose	Verify that if the encoding allows the linear element used for locating CrossSections to be an Alignment, then that encoding supports the Alignment Requirements Class.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

## A.7 Conformance class: Railway

<b>/conf/railway</b>		
Requirements	<a href="#">/req/railway</a>	
Dependency	<a href="#">/conf/facility</a>	
Test	<b>/conf/railway/classes</b>	
	Requirement	<a href="#">/req/railway/classes</a>
	Test purpose	Verify that the encoding provides the Railway Requirements Class Classes shown in blue in Figure 36 in a manner consistent with the encoding.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b>/conf/railway/alignment</b>	
	Requirement	<a href="#">/req/railway/alignment</a>
	Test purpose	Verify that if the encoding allows the linear element used for locating RailwayElements or CantEvents to be an Alignment, then that encoding supports the Alignment Requirements Class.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b>/conf/railway/cant</b>	
	Requirement	<a href="#">/req/railway/cant</a>
	Test purpose	Verify that the encoding specifies as a requirement that each CantSpecification shall cover the entire linear element the CantSpecification is located along from the linear element's start to its end, and that the CantSpecification shall include CantEvents at the start and end of the linear element and wherever else there is a cant break along the linear element.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

## A.8 Conformance class: Survey

<b><a href="#">/conf/survey</a></b>		
Requirements	<a href="#">/req/survey</a>	
Dependency	<b><a href="#">/conf/land-infra</a></b>	
Test	<b><a href="#">/conf/survey/classes</a></b>	
	Requirement	<a href="#">/req/survey/classes</a>
	Test purpose	Verify that the encoding provides the Survey Requirements Class Classes shown in blue in Figure 41 in a manner consistent with the encoding.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

## A.9 Conformance class: Observations

<b>/conf/observations</b>		
Requirements	<b>/req/observations</b>	
Dependency	<b>/conf/survey</b>	
<b>Test</b>	<b>/conf/observations/classes</b>	
	Requirement	<b>/req/observations/classes</b>
	Test purpose	Verify that the encoding provides the Observation Requirements Class Classes shown in blue in Figure 43 in a manner consistent with the encoding. An encoding shall decide which SurveyObservation types it will support and define appropriate requirements classes accordingly. This then would include any dependent types (example SatelliteSystemType has only to be supported if the encoding will support GNSS Observations).
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

## A.10 Conformance class: Equipment

<b>/conf/equipment</b>		
Requirements	<b>/req/equipment</b>	
Dependency	<b>/conf/survey</b>	
<b>Test</b>	<b>/conf/equipment/classes</b>	
	Requirement	<b>/req/equipment/classes</b>
	Test purpose	Verify that the encoding provides the Equipment Requirements Class Classes shown in blue in Figure 48 in a manner consistent with the encoding. An encoding shall decide which SurveySensor types it will support and define appropriate requirements classes accordingly.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
<b>Test</b>	<b>/conf/equipment/observation-correction</b>	
	Requirement	<b>/req/equipment/observation-correction</b>
	Test purpose	Verify that the encoding provides the ObservationCorrection Classes in blue in Figure 51 in a manner consistent with the encoding. The encoding shall decide which correction types it will support and define appropriate requirements classes accordingly.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

### A.11 Conformance class: SurveyResults

<b>/conf/survey-results</b>		
Requirements	<b>/req/survey-results</b>	
Dependency	<b>/conf/survey</b>	
<b>Test</b>	<b>/conf/survey-results/classes</b>	
	Requirement	<b>/req/ survey-results /classes</b>
	Test purpose	Verify that the encoding provides the SurveyResults Requirements Class Classes shown in blue in Figure 52 in a manner consistent with the encoding. An encoding shall decide which SurveyResult types it will support and define appropriate requirements classes accordingly.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
<b>Test</b>	<b>/conf/survey-results/topic-20</b>	
	Test purpose	Verify that, if the encoding supports survey results subjects, then the encoding shall support the SF_SamplingFeature and related Classes specified in the OGC Abstract Specification Topic 20, Observation and Measurements shown in Figure 54 that are appropriate to the supported subject area(s).
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

## A.12 Conformance class: LandFeature

<b>/conf/land-feature</b>		
Requirements	<a href="#">/req/land-feature</a>	
Dependency	<a href="#">/conf/land-infra</a>	
Test	<b><a href="#">/conf/land-feature/classes</a></b>	
	Requirement	<a href="#">/req/land-feature/classes</a>
	Test purpose	Verify that the encoding provides the LandFeature Requirements Class Classes shown in blue in Figure 55 in a manner consistent with the encoding.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability
Test	<b><a href="#">/conf/land-feature/alignment</a></b>	
	Requirement	<a href="#">/req/land-feature/alignment</a>
	Test purpose	If an encoding allows the linear element used for a LandLayer to be an Alignment, then verify that the encoding supports the Alignment Requirements class.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

### A.13 Conformance class: LandDivision

<b>/conf/land-division</b>		
Requirements	<a href="#">/req/land-division</a>	
Dependency	<a href="#">/conf/land-infra</a>	
Test	<a href="#">/conf/land-division/classes</a>	
	Requirement	<a href="#">/req/land-division/classes</a>
	Test purpose	Verify that the encoding provides the LandDivision Requirements Class Classes shown in blue in Figure 60 in a manner consistent with the encoding.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

#### A.14 Conformance class: Condominium

<b>/conf/condominium</b>		
Requirements	<a href="#">/req/condominium</a>	
Dependency	<a href="#">/conf/land-division</a>	
Test	<a href="#">/conf/condominium/classes</a>	
	Requirement	<a href="#">/req/condominium/classes</a>
	Test purpose	Verify that the encoding provides the Condominium Requirements Class Classes shown in blue in Figure 68 in a manner consistent with the encoding.
	Test method	Inspect the encoding to verify the above requirement.
	Test type	Capability

## Annex B Use Cases (informative)

The following use cases were developed to help understand the requirements for InfraGML and to communicate the scope and context. The conceptual model is based on these use cases.

### B.1 Alignment / Road

#### B.1.1 Create, store, present Simple Horizontal Alignment

**Name:** A-10-cspSHA

**Version:** 0.1

**Author:** Paul Scarponcini

**Date:** 20140820

**Description:** Initial concept(s) for a facility improvement alternative. Horizontal only (straight linestrings) with existing ground as Vertical. May be created by sketch software, CAD, or GIS but probably not civil design software. Often combined with GIS (wetlands, etc.).

**Goal:** Eliminate obviously bad alternatives.

**Actors:** Planners, Asset Managers

**Stakeholders:** Facility owner, Public (travelers and land owners), government organizations

#### **Basic Course:**

1. Identify need
2. Identify constraints (financing, ROW, environmental)
3. Review existing situation
4. Propose possible alternative solutions
5. Advance reasonable alternatives

**Trigger:** Unsatisfied transportation need; safety improvement

**Precondition:** Need for a transportation improvement has surfaced

**Postcondition:** Proposed improvements have been researched and documented, planning alignments have been developed

**Notes:** Because of the minimal precision requirements and the limitations of the software involved, a simple, horizontal, linestring approximation of the alignment is sufficient.

### **B.1.2 Create, modify, store, present Preliminary Design Alignment alternatives**

**Name:** A-20-cmspPDA

**Version:** 0.1

**Author:** Paul Scarponcini

**Date:** 20140820

**Description:** Preliminary Design Alignment alternatives are characterized by Horizontal focus with some circular curves, limited Vertical in selected, problem areas, simple linear referencing, and approximate overall roadway width. (Raster) map backgrounds and 3D visualization presentation possible.

**Goal:** Narrow down alternatives and determine right of way requirements.

**Actors:** Roadway/rail design engineers, other design engineers (bridge, drainage, traffic, soils, etc.), ROW department, surveyors

**Stakeholders:** Facility owner, Public (travelers and land owners), government organizations, utility companies

#### **Basic Course:**

1. Review inputs
  - a. proposed improvements
  - b. planning alignments (from A-cspSHA)
  - c. existing ROW
  - d. constraints (financing, ROW, environmental)
  - e. previous designs
  - f. constraints from other disciplines
  - g. survey input
2. Create preliminary road design alternative(s)
  - a. create preliminary road layout
  - b. create preliminary road Horizontal alignment

- c. create preliminary road Vertical alignment
  - d. create preliminary road 3D models
  - e. create preliminary road raster layouts
3. Share preliminary road design with other disciplines
  4. Evaluate roadway alternatives
    - a. Determine, document environmental impact
    - b. Compare costs
    - c. Determine ROW requirements
  5. Revise, approve roadway alternatives
  6. Document, present preliminary design

**Trigger:** Approval of and funding for proposed improvements.

**Precondition:** Proposed improvements have been documented, planning alignments have been provided, and funding has been identified

**Postcondition:** Preliminary design alternative is selected and preliminary design alignment and ROW requirements are documented

**Notes:** The preliminary design may be tabled until funding and/or ROW are acquired or else detailed design may commence immediately. Detailed design may be undertaken by the preliminary designers, a new design team, or by a design-build team.

### **B.1.3 Create, modify, store, present Detailed Design Alignment**

**Name:** A-30-cmspDDA

**Version:** 0.1

**Author:** Paul Scarponcini

**Date:** 20140820

**Description:** Detailed Design Alignment is characterized by Horizontal (x,y) which can contain lines, circular curves, and transition curves (clothoids for roadway) and Vertical (location along, z) which can contain symmetric or asymmetric parabolic or circular vertical curves. These can be combined into 3D strings. The Detailed Design Alignment can be combined with ground profiles, superelevation, and templates to get cross sections, 3D strings, solids, and earthwork volumes.

**Goal:** Provide exact location/geometry for the construction company to build

**Actors:** Roadway/rail design engineers, other design engineers (bridge, drainage, traffic, soils, etc.), ROW department, surveyors

**Stakeholders:** Facility owner, Public (travelers and land owners), government organizations, utility companies

#### **Basic Course:**

1. Review preliminary design outputs from roadway and other disciplines (e.g., bridge)
2. Gather additional input information
  - a. ROW constraints, access requirements
  - b. survey: location information, DTM, aerial
  - c. utility information, plans
  - d. design constraints, policies, and procedures
  - e. traffic volumes, loads
  - f. soils information, borings
  - g. other stakeholder requirements

3. Create detailed road design
  - a. develop top surface roadway design
    - i. review, revise road horizontal alignment
    - ii. develop road geometrics
    - iii. create road vertical profile
    - iv. create road (top surface) typicals
    - v. design road superelevation
    - vi. develop road (top surface) cross sections
    - vii. determine prelim (top surf) earthwork
    - viii. iterate top surface road design
    - ix. modify construction limits
  - b. add subsurface roadway design
    - i. provide soils/pavement rec
    - ii. complete road x sects w/ subsurface
    - iii. complete road typicals w/ subsurface
    - iv. determine final earthwork
    - v. iterate road design
  - c. perform miscellaneous road detail design
  - d. determine quantities
  - e. create final roadway plans
  - f. prepare road special provisions
  - g. develop road project stage construction
4. Coordinate detailed design with other disciplines
5. Develop engineer's estimate
6. Conduct design review with stakeholders

7. Submit plans, specifications, and estimates
8. Archive project design

**Extensions:** For Design-Build contracts, this use case is extended by A-40-cspuCA.

**Trigger:** Preliminary Design is approved and funding is available.

**Precondition:** Preliminary Design alternative is selected.

**Postcondition:** Detailed design is complete and ready for construction.

**Notes:** Flexibility in specifying location along the Horizontal alignment is accommodated with the specification of a Linear Referencing Method (such as, but not limited to, stationing).

#### **B.1.4 Create, store, publish, utilize Construction Alignment**

**Name:** A-40-cspuCA

**Version:** 0.1

**Author:** Paul Scarponcini

**Date:** 20140820

**Description:** Construction Alignment is the construction company's view of the Detailed Design Alignment used to construct the project. It may be in a proprietary format for use by the construction company's application software. The information transmitted to the site for surveying and machine control may be standardized or may be provided by a direct software link.

**Goal:** Provide exact location/geometry for use in construction

**Actors:** Construction company, surveyors, inspectors, Roadway/rail design engineers, other design engineers (bridge, drainage, traffic, soils, etc.)

**Stakeholders:** Facility owner, Public (travelers and land owners), government organizations, utility companies

#### **Basic Course:**

1. Review detailed engineering design plans and specifications
2. Create construction model
3. Provide bid in response to tender
4. Gather additional input information
  - a. survey: location information, DTM, aerial
  - b. utility information, plans
  - c. soils information, borings
  - d. other stakeholder requirements
5. Construct project improvements
6. Monitor construction progress

## 7. Document design changes

**Extensions:** For Design-Build contracts, this use case extends A-30-cmspDDA. Tendering occurs after Preliminary Design, as the Design-Build Contractor creates the Detailed Design outlined in A-30-cmspDDA.

**Trigger:** Decision to proceed with construction.

**Precondition:** Detailed Design is approved, funding is available, and ROW has been obtained.

**Postcondition:** Construction is complete and ready for service.

**Notes:**

### **B.1.5 Create, store As-Built Alignment**

**Name:** A-50-csABA

**Version:** 0.1

**Author:** Paul Scarponcini

**Date:** 20140820

**Description:** Changes to the as-designed Detailed Design Alignment (use case A-30-cmspDDA) may have been made during construction. The Detailed Design Alignment is updated to reflect these changes and stored as the As-Built Alignment.

**Goal:** Create a record of what was actually constructed.

**Actors:** Construction company, surveyors, inspectors, Roadway/rail design engineers, other design engineers (bridge, drainage, traffic, soils, etc.), ROW department, utility companies

**Stakeholders:** Facility owner, Public (travelers and land owners), government organizations,

#### **Basic Course:**

1. Review design changes made during construction
2. Revise Detailed Engineering Alignment accordingly
3. Archive project file

**Trigger:** Construction completion.

**Precondition:** Changes have been made to the Detailed Design Alignment during construction.

**Postcondition:** A permanent record of the As-Built Alignment exists.

**Notes:**

### **B.1.6 Create, store, publish In-Service Alignment(s)**

**Name:** A-60-cspISA

**Version:** 0.1

**Author:** Paul Scarponcini

**Date:** 20140820

**Description:** The As-Built Alignment is used to generate whatever versions of the alignment as may be necessary once the facility is in service. These alignments provide the geometric and location reference line used with linear referencing to link in other information along the facility.

**Goal:** Create alignment views necessary to support Operations, Maintenance, and Asset Management applications and databases.

**Actors:** Facility owners, asset managers, operators, maintenance managers, surveyors

**Stakeholders:** Public (travelers and land owners), government organizations, utility companies

#### **Basic Course:**

1. Retrieve As-Built Alignment
2. Create and maintain In-Service Alignment(s) as required, such as
  - a. Pavement management: linestring representation with reference posts.
  - b. Hydrology uses engineering alignment with stationing.
  - c. Survey keeps as-built alignment for future work.
  - d. HPMS uses reference posts.

**Trigger:** Facility is in service.

**Precondition:** As-Built Alignment exists. Otherwise the as-designed Detailed Design Alignment must be updated, possibly by field inspection.

**Postcondition:** In-Service Alignment(s) are available to support other applications and databases.

**Notes:** Various Linear Referencing Methods may be employed against the In-Service Alignments, based upon the specific requirement of each application/database.

## **B.2 Survey**

### **B.2.1 S-1 Generate Control Points from Survey raw measurements**

**Name:** Generate Control Points from Survey raw measurements

**Description:** Multiple survey field software applications may be used during the site selection and location planning. The collected raw measurements are later used to generate new control points after running through adjustments or specific defined methods. This is necessary when the application requires multiple measurements for the determination of the control points. Example would be running a Traverse in the field – adjust the Traverse in the office.

**Goal:** It should be possible for any survey office software applications to import all raw data measurements from an InfraGML document that was exported by any field survey software application.

**Actors:** Civil engineer, Cadaster experts

#### **Stakeholders:**

1. Designers, Owners, Cadaster, Management company

#### **Basic Course:**

1. A survey field software application exports any raw data measurements as an InfraGML document.
2. Another software application with an adjustment kernel imports all the raw data measurements from the InfraGML document.
3. The software application creates new control data out of the raw data

**Notes:** Currently with LandXML not all of the required information about the measurements are included in the data. Important information regarding sensor calibrations and quality are missing. Additionally, observation groups, target definitions and setup information could be written in different ways.

**Post Conditions:** Seamless data flows between Survey Field Software and software with a surveying adjustment kernel.

### **B.2.2 S-2 Interoperability between survey field systems and survey office systems**

**Name:** Interoperability between survey field systems and survey office systems

**Description:** Multiple survey field software applications may be used during the whole life cycle of a project. The collected measurements must be controlled or maybe corrected in a surveying office package. This could be a stand-alone survey package as middle ware or a survey plugin inside a CAD software package.

**Goal:** It should be possible for surveying software applications to import any raw data measurements from an InfraGML document that was exported by any field survey software application.

**Actors:** Civil engineer, Surveyor

#### **Stakeholders:**

1. Designers, Owners, Management company, Builder

#### **Basic Course:**

1. A survey field software application exports any raw data measurements as an InfraGML document.
2. Another software application imports all the raw data measurements from the InfraGML document and control or correct the measurements. (Field errors)

**Notes:** Currently a lot of different survey packages are used to for controlling / correction of the data. LandXML could not manage for e example different sets of reflectors etc. – there are different meddle ware in the market and most of the survey application packages still uses different formats (fbk, rw5..)

**Post Conditions:** Seamless data flows between Survey Field Software and other survey software to control ant correct measurements.

### **B.2.3 S-3 InfraGML is used as fieldbook reports**

**Name:** InfraGML is used as fieldbook reports

**Description:** Multiple survey field software applications may be used during the whole life cycle of a project. The fieldwork must be reported and archived. InfraGML could be used to hold all the raw data and with the use of Style Sheets to transform in in the required customized reporting form.

**Goal:** It should be possible to hold all the field work in an InfraGML document that could be used as a fieldbook.

**Actors:** Surveyor

#### **Stakeholders:**

1. Designers, Owners, Management company, Builder

#### **Basic Course:**

1. A survey field software application exports any raw data measurements as an InfraGML document.
2. Another software application imports all the data from the InfraGML document and transform it into the required reporting format.

**Notes:** Currently a lot of different reporting formats are locally developed by each surveying field software packages.

**Post Conditions:** Style Sheets and InfraGML could be used for archive and reporting survey field works.

**Post Conditions:** Style Sheets and InfraGML could be used for archive and reporting survey field works.

#### **B.2.4 S-4 New sensor supports**

**Name:** New sensor supports

**Description:** In the last years more and more new sensors could be used during a project. For example, high definition laser scanning, terrestrial photogrammetry, terrestrial positions and GNSS are combined in a modern survey instrument.

**Goal:** It should be possible to combine all the different survey technologies inside an InfraGML and export all the collected data. The office package should have access to all the different technologies and take advantages of each system dependent of the application.

**Actors:** Civil Engineer

#### **Stakeholders:**

1. Designers, Owners, Management company, Builder

#### **Basic Course:**

1. A survey field software application exports any raw data measurements of the different measurement systems as an InfraGML document.
2. Another software application imports all the data from the InfraGML document.

**Notes:** Combination of the different technologies in one package is not yet available. More and more packages are combining these technologies.

**Post Conditions:** Software packages that could manage high definition laser scanning, terrestrial photogrammetry, terrestrial positions and GNSS.

### **B.2.5 S-5 Interoperability between survey field systems and monitoring systems**

**Name:** Interoperability between survey field systems and monitoring systems

**Description:** Multiple survey field software applications for monitoring objects. The collected raw measurements are later used in specific monitoring software applications.

**Goal:** It should be possible for monitoring software applications to import any raw data measurements from an InfraGML document that was exported by and field survey software application.

**Actors:** Civil engineer, Monitoring engineer

**Stakeholders:**

1. Owners, Management company

**Basic Course:**

1. A survey field software application exports any raw data measurements as an InfraGML document.
2. Another monitoring software application imports all the raw data measurements from the InfraGML document.

**Notes:** Currently with LandXML not all of the required information about the measurements are included in the data. Important information regarding sensor calibrations and quality are missing. Additionally, observation groups, target definitions, and setup information could be written in different ways.

**Post Conditions:** Seamless data flows between Survey Field Software and monitoring software.

## **B.3 General**

### **B.3.1 G-1 Porting old LandXML data to InfraGML**

**Name:** Porting old LandXML data to InfraGML

**Version:** 0.1

**Author:** Julian Amann

**Date:** 20140303

**Description:** A user wants to convert his old LandXML data to the new InfraGML data model.

**Goal:** It should be possible to store the LandXML data as an InfraGML file.

**Actors:** Civil engineer.

**Stakeholders:** User who wants to store data as an InfraGML file.

#### **Basic Course:**

1. We have an existing LandXML file that contains data
2. There is a mapping between the data of LandXML and InfraGML

**Precondition:** The LandXML file is a validate LandXML document.

**Notes:** For alignments, for example, InfraGML must be able to support all alignment descriptions of LandXML. This means it must support in some way vertical and horizontal alignments. It must support pure 3d alignments (irregular line in LandXML).

### **B.3.2 G-2 Porting InfraGML data back to LandXML**

**Name:** Porting InfraGML data back to LandXML

**Version:** 0.1

**Author:** Julian Amann

**Date:** 20140303

**Description:** A user wants to convert new InfraGML data model into LandXML for use in legacy LandXML applications.

**Goal:** It should be possible to store the InfraGML data as a LandXML file.

**Actors:** Civil engineer.

**Stakeholders:** User who wants to use InfraGML data in applications that only support LandXML.

**Basic Course:**

1. We have an existing InfraGML file that contains data
2. There is a mapping between the data of LandXML and InfraGML

**Precondition:** The InfraGML file is a validate InfraGML document.

**Notes:**

### **B.3.3 G-3 Digital preservation**

**Name:** Digital preservation.

**Version:** 0.1

**Author:** Julian Amann

**Date:** 20140303

**Description:** User wants to store data for a long time.

**Goal:** For maintenance reasons or future projects the InfraGML data should be archived.

**Actors:** Engineer.

**Stakeholders:** Government, Contractor

#### **Basic Course:**

1. User creates an InfraGML document
2. User wants to archive it

**Notes:** The OGC Preservation DWG should be consulted to see if they have established requirements or procedures for this.

## **B.4 Terrain**

### **B.4.1 T-1 Store terrain data**

**Name:** Store terrain data

**Version:** 0.2

**Author:** Paul Scarponcini, Julian Amann

**Date:** 20141216

**Description:** It should possible to store the surface of a digital elevation model as well as subsurface soils layers bounded by surfaces.

**Goal:** Be able to exchange existing and proposed terrain data amongst designers.

**Actors:** Site engineers, other design engineers (roadway/rail, bridge, drainage, soils, etc.), surveyors.

**Stakeholders:** Facility owner, land owner, government organizations, utility companies

#### **Basic Course:**

1. Terrain is measured at different positions. For each position a height value is determined.
2. Terrain data needs to be stored, either as height data or as a TIN, or both.
3. Subsurface data is collected, usually by soil borings.
4. Layers can be defined between surfaces or as 3D solids (e.g., Polyface meshes).
5. As design work progresses, new proposed terrain surfaces can be defined and differences between existing and proposed can be determined as earthwork cut and fill quantities.

## **B.5 Land Division**

The following use cases reflect the scope of LandInfra which include interchange of survey plan data in terms of documents to be lodged at national recording agencies, but excludes management of land recordings and database storage; cf. the introduction of 7.10 LandDivision

### **B.5.1 P-1 Right of way (ROW) for infrastructure project**

**Name:** Right of way (ROW) for infrastructure project

**Version:** 0.3

**Author:** Paul Scarponcini and Erik Stubkjær

**Date:** 20150714

**Description:** Provision of right of way for infrastructure improvement projects, including cadastral survey and recording of the completed project

**Goal:** Provision of right of way for infrastructure project

**Actors:** Infrastructure department, ROW department, owners, surveyor, land agency, government

**Stakeholders:** Neighbours, government organizations (land registry, tax, heritage protection), utility companies, SDI technology providers

#### **Basic Course:**

1. Infrastructure department prepares project proposal, including possible corridors
2. ROW department ascertains legal and economic preconditions for project:
  - need of parcel acquisition and easement forming
  - need of access re-routing, due to new infrastructure
  - compensation for temporary and permanent damage
  - compliance with restrictions due to land use plans, heritage protection, etc.
3. Government assesses functional and economic implications and allocates resources
4. ROW department peg out infrastructure boundaries, inform owners of implications, and note owner demands
5. Project proposal is finalized in terms of infrastructure location and compensation to owners

- ROW department establishes ROW through Real Property Acquisition, as well as Easement Forming. (Corresponding use cases are not specified).
6. Infrastructure department arranges for infrastructure construction
  7. ROW department reports as-built infrastructure boundaries to land agency through modified UseCase\_LandParcelChange

### References:

- Leif Norell (2007) Markåtkomst och ersättning för vägar, järnvägar och kraftledningar i Norden - En översikt över processerna i Danmark, Finland, Island, Norge och Sverige (Acquisition of land and compensation for roads, railroads and powerlines in the Nordic countries - A survey of the processes in Denmark, Finland, Iceland, Norway and Sweden) - <http://gst.dk/media/gst/2626920/Nordiskjämförelsesluttrapport.pdf>

### Note:

Infrastructure department, ROW department, and land agency are abstract, organizational entities, which are casted for the purpose of describing a sufficiently generic use case. National law describes the specific mandate and operation of these entities.

ROW department is assumed to be staffed with geodetic surveying and legal expertise, in order to ascertain the protection of individual property rights. Expropriation / compulsory purchase procedures, typically performed by an independent, quasi-judicial body, are similarly implied, but not articulated due to the applied level of abstraction.

### **B.5.2 P-2 Pointing out an existing real property boundary (Resurvey)**

**Name:** Pointing out an existing real property boundary (Resurvey)

**Version:** 0.2

**Author:** Erik Stubkjær

**Date:** 20150714

**Description:** The existing legal boundary line is pointed out, typically as a prerequisite for subsequent construction or land development activities

**Goal:** The demonstration on location of a legal boundary line.

**Actors:** Owner, surveyor, land agency

**Stakeholders:** Neighbors

#### **Basic Course:**

1. Owner requests surveyor assistance for pointing out an existing real property boundary
2. Surveyor assesses legal and financial implications and takes on the case
3. Surveyor collects / queries evidence concerning the case from owner and land agency
4. Surveyor identifies in the collected evidence information (points), which refer to presumably stable and well-defined locations (points) in the terrain, including monumented points
5. Surveyor transforms evidence in terms of {traverse and offset measurements | measurements in Cartesian coordinate systems} into a common coordinate system, to establish a consolidated set of boundary point coordinates
6. Surveyor sets out the consolidated set of point coordinates {on location | in the field}, compares the set out points with the location of {boundary marks | monuments} and other terrain features, e.g. fences, hedge rows and walls, and searches for perhaps hidden monuments.
7. Surveyor collects oral evidence from owner and neighbors, e.g. on hedge age and maintaining practices, and - if allowed by the so provided evidence - points out the legal boundary.
8. Surveyor monuments (selected) legal boundary points and prepares a measurement sheet, showing the legal boundary a stable, well-defined terrain features in the vicinity, including building corners and monumented geodetic points.
9. Surveyor hands over documentation to owner in return for remuneration (honorarium, fees, etc.)

#### **Variants:**

Steps 5 and 7-8.

If the collected evidence consists of measurements in the same geographic coordinate system, covering the case and referring to monumented boundary points, steps 5.-8. are correspondingly reduced or omitted.

### Step 7

If the evidence available, e.g. conflicting neighbor claims, does not allow for pointing out the legal boundary, national law advises next steps.

### References:

[https://en.wikipedia.org/wiki/Cadastral\\_surveying](https://en.wikipedia.org/wiki/Cadastral_surveying) (the term: Resurvey)

### **B.5.3 P-3 Land parcel development /subdivision / change**

**Name:** Land parcel development /subdivision / change

**Version:** 0.1

**Author:** Erik Stubkjær

**Date:** 20150622

**Description:** Forming of new parcels or relocation of existing boundaries

**Goal:** Protection of real property rights through parcel identification, boundary measurement, and public recording

**Actors:** Owner, surveyor, cadaster / land administration agency, local government

**Stakeholders:** Neighbors, government organizations (land registry, tax, heritage protection, etc.), SDI technology providers

#### **Basic Course:**

1. Owner requests surveyor assistance for change of property boundaries
2. Surveyor assesses legal and financial implications and takes on the case
3. Surveyor collects public recorded evidence concerning the case
4. Surveyor ascertains legal preconditions for change
  - o submits application(s) for approval concerning land use, heritage protection, etc. restrictions
  - o arranges with right holders concerning easements and mortgages
5. Surveyor identifies and measures 'connection / reference points', determines existing boundaries on location, and sets out, marks and measures new boundaries.
6. Surveyor identifies temporary parcel lots and future parcels and accounts for area and boundary changes (by means of survey sheet and parcel account sheet)
7. Surveyor submits change documentation to local government for approval
8. Surveyor submits change documentation to cadaster agency for approval and recording
9. Cadaster agency informs surveyor and government organizations of the changed records
10. Surveyor hands over documentation to owner in return for remuneration (honorary, fees, etc.)

#### **References:**

- Stubkjær, E. (2002) Modelling Real Property Transactions. Paper presented at the XXII FIG Congress, Washington, D.C. USA, April 19-26 2002
- Property formation in the Nordic countries - Denmark. English translation of the introduction, Danish chapter and comparison. 2006
- MBBL, Ejendomsdataprogrammet: GD1 - Løsningsarkitektur - Bilag C Processer 0.8.docx

#### **B.5.4 P-4 Forming strata title / condominium**

**Name:** Forming strata title / condominium

**Version:** 0.1

**Author:** Erik Stubkjær

**Date:** 20150622

**Description:** Forming strata title units in a multi-storage building

**Goal:** Protection of property rights through condominium identification, boundary measurement, and public recording

**Actors:** Owner, surveyor, cadaster / land administration agency, local government

**Stakeholders:** Neighbors, government organizations (land registry, tax, heritage protection), SDI technology providers

#### **Basic Course:**

1. Owner requests surveyor assistance for forming of condominium units
2. Surveyor assesses legal and financial implications and takes on the case
3. Surveyor collects public recorded evidence concerning the case
4. Surveyor ascertains legal preconditions for change
  - o by-laws for the owner association prepared for recording
  - o principles for share attribution and unit formation established
5. Surveyor identifies and measures 'connection / reference points' and existing boundaries, measures the footprint of the building, and floor / story-wise the boundaries of condominium units and shared facilities.
6. Surveyor identifies condominium units and accounts for each unit its area and share in common property (by means of survey sheets and condominium account sheet)
7. Surveyor submits change documentation to land registry / land administration agency for recording
8. Surveyor hands over documentation to owner in return for remuneration (honorary, fees, etc.)

## **B.6 Railway**

### **B.6.1 R-10 Handling chainage discontinuities between tracks**

**Name:** Handling chainage discontinuities between tracks

**Version:** 0.1

**Author:** Peter Axelsson

**Date:** 20151021

**Description:** Rail facilities with direct reference to the track are often referred to via chainage or kilometrage data relating to them. Each track or route is given its own chainage or kilometrage in case of the route. Chainage discontinuity is the modified chainage of a track or route caused by modification of the track or route length.

**Goal:** An implementation in e.g. GML or GeoPackage should be able to handle the chainage discontinuities for calculations of true distances along the track. Both extensions and shortenings of tracks must be handled.

**Actors:** Construction company, surveyors, inspectors, rail design engineers

**Stakeholders:**

**Basic Course:**

1. Export/import of existing tracks/alignments and chainage points and values
2. Redesign and reconstruction of a track that modifies the length of the track
3. Calculation of new connection/chainage points and values
4. Export/import of new alignments and chainage points and values

**Trigger:** Changing the alignment of a track

**Postcondition:** New values and points for chainage discontinuities are calculated and stored.

**Notes:**

## **B.6.2 R-11 Conversion between representations of positions**

**Name:** Conversion between representations of positions in the forms of Cartesian coordinates, true distances along track and local kilometrage distance

**Version:** 0.1

**Author:** Peter Axelsson

**Date:** 20151021

**Description:** When multiple alignments/tracks with chainage discontinuities are connected there is a need to convert between the different forms of representations, such as Cartesian coordinates, true distance along track and local kilometrage distance.

**Goal:** An implementation in e.g. GML or GeoPackage should be able to handle the tracks, alignments, chainage discontinuities and kilometrage in such a way that conversions can be made between the different representations.

**Actors:** Construction company, surveyors, inspectors, rail design engineers

**Stakeholders:**

**Basic Course:**

1. Multiple connected alignments with information of kilometrage, true starting points and distance/true end point.
2. Position on the track in the form of coordinates, true distance or local kilometrage distance.
3. Convert the position to the other forms of representation.

**Trigger:** Positioning of events or positioning along track.

**Postcondition:**

### **B.6.3 R-20 Create, modify, store, present single objects along track**

**Name:** Create, modify, store, present single objects along track

**Version:** 0.1

**Author:** Peter Axelsson

**Date:** 20151021

**Description:** New objects along the track is created and connected to the track/alignment. The object can be positioned by its coordinates, true distance or kilometrage. The object can be on or off the track by a distance. The object can be a point object, linear object, area or volume.

**Goal:** An implementation in e.g. GML or GeoPackage should be able to position objects both by its coordinates, true distance or kilometrage.

**Actors:** Construction company, surveyors, inspectors, rail design engineers

**Stakeholders:**

**Basic Course:**

1. An object is created and connected to an alignment.
2. The position can be represented by its coordinates, true distance or kilometrage.
3. The object and positioning element can be exported to a GML or GeoPackage implementation.

**Trigger:** Creating new connected objects.

**Postcondition:**

#### **B.6.4 R-30 Create, modify, store and present multiple objects along track**

**Name:** Create, modify, store and present multiple objects along track

**Version:** 0.1

**Author:** Peter Axelsson

**Date:** 20151021

**Description:** A new group of connected objects along the track, e.g. the definition points of a switch, is created and connected to the alignment. The objects can be positioned as a group by its coordinates, true distance or kilometrage. The objects can be on or off the track by a distance. The object can be point objects, linear objects, areas or volumes.

**Goal:** An implementation in e.g. GML or GeoPackage should be able to position groups of objects both by its coordinates, true distance or kilometrage.

**Actors:** Construction company, surveyors, inspectors, rail design engineers

**Stakeholders:**

**Basic Course:**

1. A group of object is created and connected to an alignment.
2. The position can be represented both by its coordinates, true distance or kilometrage.
3. The group of objects and positioning element can be exported to a GML or GeoPackage implementation.

**Trigger:** Creating new group of connected objects.

**Postcondition:**

### **B.6.5 R-40 Create, modify, store, present cant**

**Name:** Create, modify, store, present cant

**Version:** 0.1

**Author:** Peter Axelsson

**Date:** 20151021

**Description:** The cant of a railway track (also referred to as superelevation) is the difference in elevation (height) between the two rails. This is normally done where the railway is curved.

**Goal:** An implementation in e.g. GML or GeoPackage should be able to handle the Cant values at defined positions along the alignment.

**Actors:** Construction company, surveyors, inspectors, rail design engineers

**Stakeholders:**

**Basic Course:**

1. Cant values (elevation and side) are introduced or changed at defined positions during design
2. Cant values can be exported to GML or GeoPackage implementation.

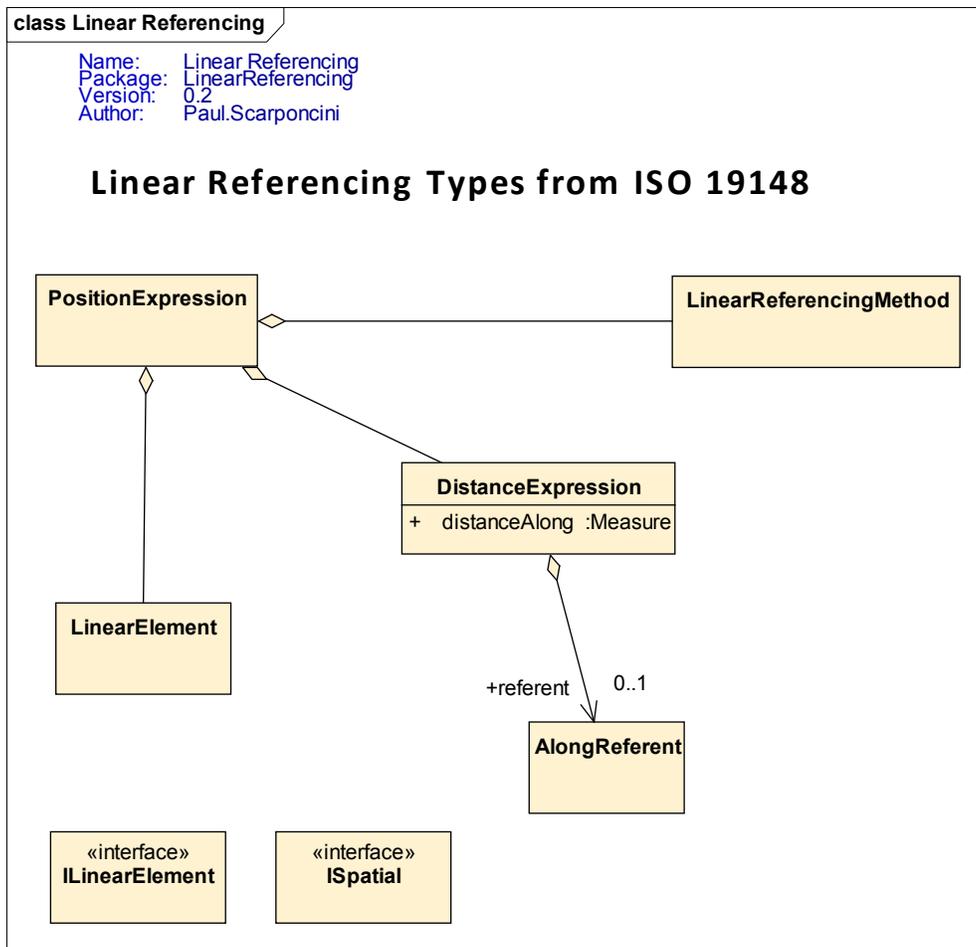
**Trigger:** New cant values added

**Postcondition:**

**Notes:**

## Annex C Linear Referencing (informative)

### C.1 ISO 19148 - Linear referencing



A PositionExpression is used to specify a linearly referenced location. In accordance with ISO 19148 (the source for OGC Abstract Specification Topic 19, Linear Referencing), It must have three components:

- 1) the linear element along which measurements are being made,
- 2) the method used to do the measuring (the Linear Referencing Method, or LRM), and

3) the actual measured value along (and optionally offset from) the linear element.

Some of these values may be implied. For example, if an Alignment has an Alignment2DVertical, it must have a single Alignment2DHorizontal along which distance along measures can be made. The Alignment2DHorizontal therefore acts as the linear element for these measured values. An Alignment also has a default LRM. It is therefore not necessary to include neither the linear element nor the LRM in the Alignment2DVertical distance along measures. This is merely a DistanceAlong measured value (decimal number and units of measure) for simple “absolute” type LRMs such as chainage and absolute stationing. For the more complex “relative” type of LRM (e.g., relative stationing with station equations), a Referent would be required.

PositionExpression - a linearly referenced location given by the linear element being measured, the method of measurement, and a measure value specified with a distance expression.

LinearElement - the underlying linear element upon which the measures in the Linear Referencing System are made.

LinearReferencingMethod - describes the manner in which measurements are made along (and optionally offset from) a linear element.

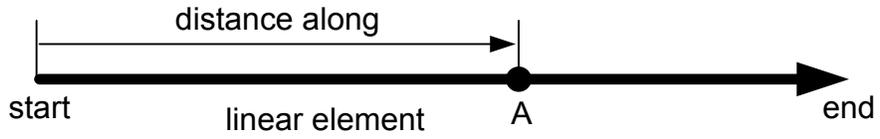
DistanceExpression - specifies the linear referenced measure value. This shall include the distance measured along the linear element. If the Linear Referencing Method LRMTType is “relative”, the distance expression may also include an along referent to specify where the measuring begins. Otherwise, measuring begins at the start of the linear element. Measuring is in the direction of the linear element, unless a towards referent is provided.

AlongReferent - for “relative” type Linear Referencing Methods, the AlongReferent is used to specify the location along the LinearElement at which the measuring is to begin (e.g., a milepost).

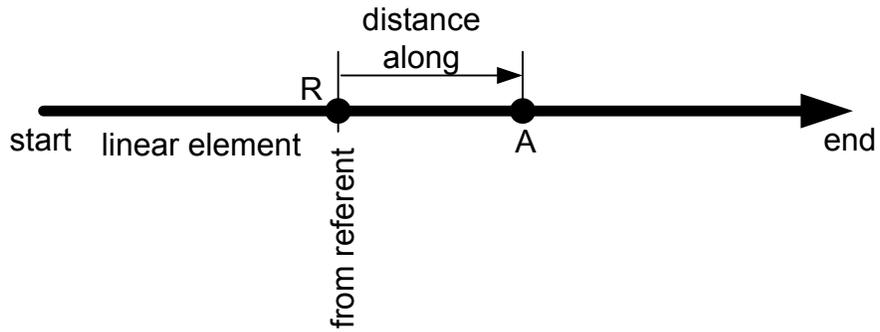
For absolute LRMs, the linearly referenced location (A) is determined by measuring a distance along the linear element from its start<sup>2</sup>:

---

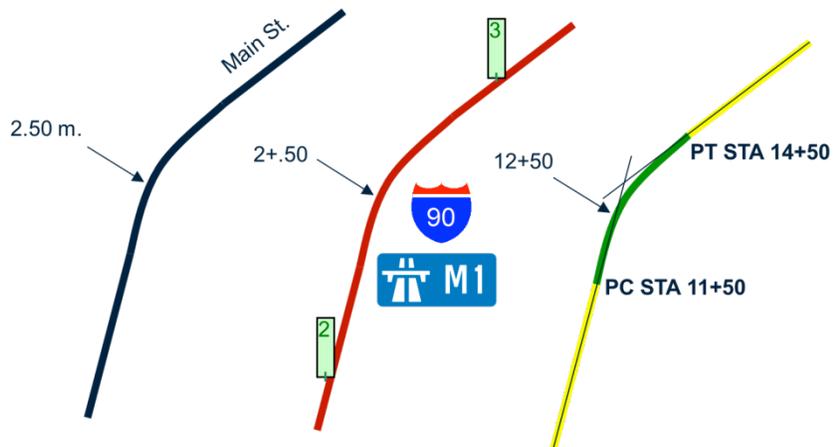
<sup>2</sup> Figures of this type are from ISO 19148:2012.



For relative LRMs, the linearly referenced location (A) is determined by measuring a distance along the linear element from a known location (R) on the linear element:



A few examples of linear referenced locations (with absolute LRMs) are depicted below.



In the first example, the linearly referenced location is 2.5 meters along Main St., measured from the start of the street. This is specified by a PositionExpression composed of Main St. as the LinearElement, chainage (in meters) as the LinearReferencingMethod, and 2.5 (meters) as the DistanceExpression.

In the second example, the linearly referenced location is .50 miles along I-90, measured from reference post 2. This is specified by a PositionExpression composed of I-90 as the LinearElement, reference post (in miles) as the LinearReferencingMethod, and a DistanceExpression containing .50 (miles) as the DistanceAlong with reference (mile) post 2 as the AlongReferent. In the UK, the PositionExpression would be composed of the M-1 Motorway as the LinearElement, reference post (with kilometers) as the LinearReferencingMethod, and a DistanceExpression containing .500 (kilometers) as the DistanceAlong with reference (kilometer) post 2 as the AlongReferent.

In the third example, the linearly referenced location is 1250 feet along the Alignment, measured from the start of the Alignment. This is specified by a PositionExpression composed of the Alignment as the LinearElement, absolute stationing (in feet) as the LinearReferencingMethod, and 1250 (feet) as the DistanceExpression.

In order to behave as a linear element, ISO 19148 requires that the linear object to be measured support the ILinearElement interface. This requires the following:

- 1) the linear element shall have a default LRM,
- 2) the full length of the linear element shall be known or obtainable,
- 3) it shall be possible to obtain a distance expression which represents the translation of a linearly referenced location on this linear element onto a specified target element using a specified Linear Referencing Method
- 4) it shall be possible to obtain the set of position expressions which represent the translation of a linearly referenced location on this linear element onto the appropriate element instances of the specified target type, using a specified Linear Referencing Method
- 5) the linear element shall have a measure value at the beginning of the linear element for a specified Linear Referencing Method

ISO 19148 also makes it possible to translate from linearly referenced locations to spatial (x, y, and maybe z) positions, or vice versa. To do this, the linear element must also support the ISpatial interface. The ISpatial interface includes operations which return:

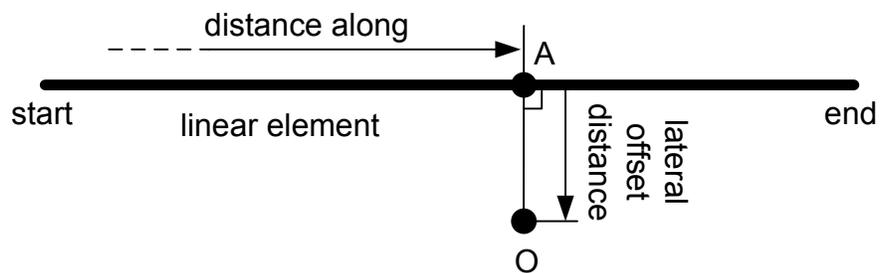
- 1) a spatial position of type Point spatially equal to the specified linearly referenced location along the subject linear element

- 2) a DistanceExpression that specifies the linearly referenced location along the subject linear element using the linear element's default Linear Referencing Method and which is spatially equal to the specified spatial position

## C.2 Linear Referencing with Offsets

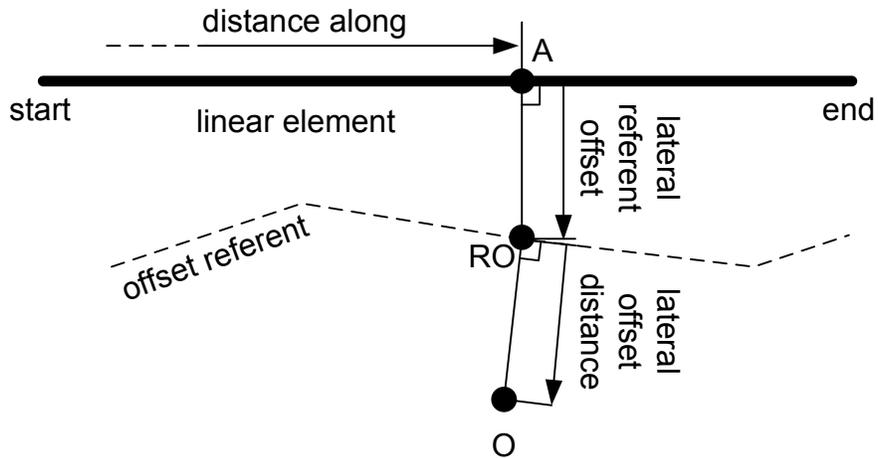
The ISO 19148 extension package, Linear Referencing Offset, supports lateral (horizontal and vertical) displacements from the distance along position for a linearly referenced location. The ISO 19148 extension package, Linear Referencing Offset Vector, supports vector offsets from the distance along position for a linearly referenced location.

The linear referenced location may not fall on the linear element but instead may be offset left or right of the linear element:



The distance along is first measured in accordance with the LRM as discussed previously to get to location A. Then the lateral offset distance is measured laterally from A in a direction perpendicular to the linear element to get to the linear referenced location O.

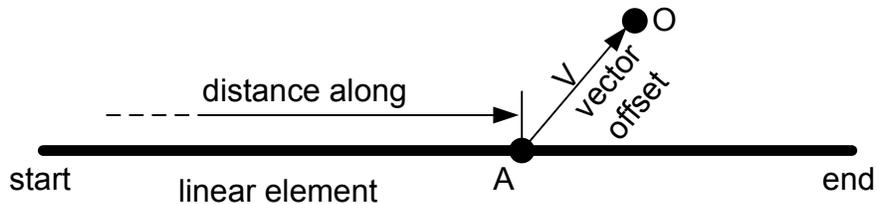
Similar to relative distance along measures, it is also possible to define linear referenced locations with relative offsets:



The offset referent can either be a character string description (e.g., back of sidewalk) or an actual geometry/location. Only in the latter case can the spatial position of O be determined, assuming the geometry/location of the linear element is also known.

Vertical offsets are handled in similar fashion. To determine the (3D) spatial position of O for both absolute and relative vertical offsets, the known linear element geometry/location must be 3D.

For even greater flexibility, offsets can be defined at any (2D or 3D) angle from the linear element, using vector offsets:



See ISO 19148 for a more detailed description of offsets.

### C.3 Alignments and Linear Referencing

In order for an Alignment to function as a linear element, it must support the `ILinearElement` interface. Requirement 1 is satisfied by the `lrm` attribute of `Alignment`. Requirement 2 can be satisfied by calculating the length of the `2dLinestringRepresentation`, the sum of the lengths of the `Alignment2DHorSegments`, or the length of the `Alignment3D`. Requirement 5 is satisfied by the `startDistantMeasureAlong` attribute of `Alignment` (zero if it is missing).

In order to support the translations specified in Requirements 3 and 4, the `Alignment` (source) and/or its LRM must be mapped to the target linear element and/or target LRM. This is typically done by equating equivalent linear referenced locations along the source and target and then using a linear interpolation algorithm to do the translation for locations in between the mapped locations.

Consider an `Alignment` as defined above in Section 5, with an absolute stationing LRM. It would be possible to convert distance along values into distances along the same `Alignment` but in accordance with a chainage LRM, if the mappings are defined. This can be as simple as defining the chainage values for the starting and ending absolute station values.

It would also be possible to translate `Alignment` absolute station linear referenced locations into equivalent chainage linear referenced location values along the corresponding link in an asset management system network model if the start and end link chainage values are mapped from/to stationed locations on the `Alignment`.

To achieve translations to and from spatial positions, the geometry/location of the linear element must be known. For `Alignments`, this is embodied in the `2dLinestringRepresentation`, the `Alignment2DHorSegments`, or the `Alignment3D`.

## Annex D Comparison with Other Standards (informative)

### D.1 LandXML

Table 3 provides a comparison between the concepts in LandInfra and LandXML 1.2 [1]. In this table, RC = Requirements class and NCC = No Corresponding Concept.

LandInfra	LandXML	Comments
<b>7.2 LandInfra RC</b>		
7.2.1.1 LandInfraDataset	LandXML element	
7.2.1.2 Feature Class	NCC	LandXML “feature” is equivalent to LandInfra “property”
7.2.1.5 Property	Feature element	User-defined property
<b>7.3 Facility RC</b>		
7.3.1.1 Facility	NCC	
<b>7.4 Project RC</b>		
7.4.1.1 Project	Project element	
<b>7.5 Alignment RC</b>		
7.5.2 Alignment Set	Alignments element	LandXML collection of horizontal alignments
7.5.3 Horizontal Alignment	Alignment element	

<b>LandInfra</b>	<b>LandXML</b>	<b>Comments</b>
NCC	Profile element	LandXML collection of existing and proposed vertical alignments
7.5.4 Vertical Alignment	ProfSurf, ProfAlign elements	
7.5.5 3D Alignment	NCC	
<b>7.6 Road RC</b>		
7.6.2 Road Element	NCC	Road elements of user-specified type
NCC	Road element	Road elements for FHWA IHSDM project
7.6.4 String Line	NCC	
7.6.6 Surface	Surface element	
7.6.8 RoadCrossSection	CrossSect element	
7.6.8.2 Cross Section Set	CrossSects element	
<b>7.7 Rail RC</b>		
7.7.3 RailwayElement	NCC	
7.7.5.1 Cant Specification	Cant element	
7.7.5.2 Cant Event	CantStation element	
<b>7.8 Survey RC</b>		
7.8.1.1 Survey	Survey element	
7.8.3.1 Setup	ObservationGroup element	

<b>LandInfra</b>	<b>LandXML</b>	<b>Comments</b>
7.8.3.2 InstrumentPoint	InstrumentPoint element	LandXML InstrumentPoint was only available under Instrument setup
7.8.3.4 SurveyObservation	NCC	
7.8.3.7 TPSObservation	RawObservation element	
7.8.3.9 GnssObservation	GPSPosition element	
7.8.3.11 GnssQuality	GPSQCInfoLevel1 and GPSQCInfoLevel2 element	
7.8.5.16 Correction	Correction element	In LandInfra corrections belongs to individual observations and not to the equipment in LandXML
7.8.5.1 Equipment	Equipment element	
7.8.5.3 SurveySensor	NCC	
7.8.7.2 TargetPoint	TargetPoint element	
7.8.7.7 AveragePoint	ReducedObservation element	LandXML reduces the Observations, in LandInfra the resulting point is represented in the average point.
<b>7.9 LandFeature RC</b>		
7.9.2 LandElement	PlanFeature element	LandXML also includes LandInfra Facility infrastructure elements here, such as signs and building footprints
7.9.3 Land Surface	Surface element	

<b>LandInfra</b>	<b>LandXML</b>	<b>Comments</b>
7.9.4 Land Layer	NCC	
<b>7.10 LandDivision RC</b>		
7.10.2.1 PropertyUnit	LocationAddress; RoadName	ISO 19160-1 Addressing expects FDIS ballot by end of 2015
7.10.2.2 LandPropertyUnit	Parcels	
7.10.2.3 LandParcel	Parcel	
7.10.3 AdministrativeDivision	AdministrativeArea	
7.10.4 Statement + landRecordingDocumentID	Title; Amendment	
7.10.5 SurveyMonument	Monument	
7.10.6 SpatialUnit	CoordGeom; Line; Curve	Applies both to LandParcel and to CondominiumUnit via BuildingPart
7.10.7 BoundingElements	CoordGeom; Line; Curve	
<b>7.11 Condominium RC</b>		
7.11.1 CondominiumUnit	Parcel  +lotEntitlements  +liabilityApportionment	
7.11.3 CondominiumBuilding	+buildingNo	
7.11.4 BuildingPart	+buildingLevelNo	
7.11.6 CondominiumScheme	NCC	Specifies condominium details

*Table 3. LandInfra / LandXML corresponding concepts*

## **D.2 ISO 19152:2012 Land Administration Domain Model (LADM)**

The conceptual schema of LADM [7] is structured into four packages with names and relations as follows:

1. Party Package: parties (people and organizations);
2. Administrative Package: basic administrative units, rights, responsibilities, and restrictions (ownership rights);
3. Spatial Unit Package: spatial units (parcels, and the legal space of buildings and utility networks); including
4. the Surveying and Representation Subpackage: spatial sources (surveying), and spatial representations (geometry and topology);

### **D.2.1 Party**

LandInfra does not render the LADM modelling of parties, groups and memberships. Parties appear in LandInfra in terms of class 7.2.1.8 Professional with attributes, and as attribute 7.10.4 signatory on statements. A statement concerning an easement may record the attribute 7.10.2.5 beneficiaryParty, e.g. a power company. Signatories may be representatives of companies or other legal persons, but these are not detailed.

The establishment of condominiums implies mandatory establishment of an association of condominium owners, but association name and bylaws are not considered surveying relevant and thus omitted. Some jurisdictions allow exclusive right for condominium owners to a parking lot on the parcel on which the subdivided building rests. To accommodate this need, the LandInfra text mentions that the owner association may appear as a legal person, establishing an easement which governs the use of the land parcel.

Government agencies are referred to only in the context of recording identifications, e.g. attributes 7.2.1.8 registration of a professional's license, 7.10.2 landRecordingDocumentID and 7.10.1.3 landRecordingId, respectively. Other parties include the attribute 7.8.5.3 calibrationBody.

### **D.2.2 Administrative units and ownership rights**

The LADM modelling of various rights, restrictions, responsibilities and mortgage is rendered through class 7.10.2 InterestInLand which distinguishes ownership in land parcel, ownership of condominium, and easement across one or more land parcels. Motivation for this approach includes that lease, mortgage and most other rights follow the boundaries of ownership, while only easements and condominiums have a boundary type of their own.

<b>LandInfra</b>	<b>LADM</b>	<b>Remarks</b>
<b>7.2 Core RC</b>		
7.2.1.3 Document	LA_Source	Specializations: 7.10.4, 7.11.6
<b>7.8. Survey RC</b>		
7.8.7.1 SurveyResults	~LA_SpatialSource	
<b>7.10 LandDivision RC</b>		
7.10.2 InterestInLand	~LA_RRR	
7.10.2.1 PropertyUnit	LA_BAUnit	Specializations: .10.2.2, 7.11.1
7.10.2.2 LandPropertyUnit		Not a CondominiumUnit
7.10.2.3 LandParcel	~LA_Parcel alias LA_SpatialUnit	Spatially defined by SpatialUnit
7.10.2.5 Easement	~LA_Restriction	Spatially defined by SpatialUnit
7.10.3 AdministrativeDivision	LA_SpatialUnitGroup	
7.10.4 Statement	LA_AdministrativeSource	
7.10.5 SurveyMonument	LA_MonumentationType	
7.10.6 SpatialUnit	~LA_SpatialUnit	
7.10.7 BoundingElements	LA_Point, BoundaryFaceString, ...	
<b>7.11 Condominium RC</b>		
7.11.1 CondominiumUnit		
7.11.3 CondominiumBuilding		

7.11.4 BuildingPart	LA_LegalSpaceBuildingUnit	Spatially defined by SpatialUnit
7.11.6 CondominiumScheme		Renders condominium details

*Table 4. LandInfra / LADM corresponding concepts*

### **D.2.3 Spatial units and 4 Surveying and spatial representations**

In LandInfra, the geometrical shape of spatial units is modelled independently from the legal status of these units. Thus, LandParcels, Easements and the BuildingParts of CondominiumUnits are all spatially represented by SpatialUnit, which convey geometrical aspects, including dimension and area, together with the class BoundingElements, which consolidates a number of spatial constructs.

### **D.2.4 Summary**

The scope of LandInfra is land development and civil engineering infrastructure facilities. The emphasis of LandInfra on infrastructure and on surveying suggests minimizing as far as possible the legal-administrative aspects of land development. This is achieved by modelling what is needed to account for the surveying related activities, including defining the legal entities, the boundary of which are measured, as well as identification of the signing parties.

The scope of LADM reflect that mapping and recording of land parcels often developed in service of the State, predominantly in terms of taxation. Standards need cross-national English technical terms, a challenge addressed by LADM. The narrower scope of LandInfra fits the notion of ‘cadastral survey’, an Anglo-American concept which relates the mapping of land parcels primarily to the securing of proprietary interest in land.

### D.3 INSPIRE Data Specification

The INSPIRE directive aims to create a European Union (EU) spatial data infrastructure [16]. The INSPIRE directive came into force in 2007 and will be implemented in various stages, with full implementation required by 2019. The Directive addresses 34 spatial data themes needed for environmental applications, with key components specified through technical implementing rules.

In the following, the Annex I themes (post) Addresses, Administrative Units, Cadastral Parcels, Coordinate Reference Systems and Geographical names, and the Annex II theme Buildings are compared with LandInfra content. The INSPIRE themes Transport networks and Utility and Government Services, respectively, relates to LandInfra, but are not addressed.

#### D.3.1 Theme Addresses

LandInfra attributes zero to more postAddresses, rendered as CharacterString, to the PropertyUnit (real property), with its subtypes CondominiumUnit and LandParcel. This compares to the [0 .. \*] cardinality between INSPIRE Address and CadastralParcel. NOTE 2 that Address may not be restricted to only a one to one relationship with cadastral parcel is thus observed. The position of the post address, as well as address components are not specifically accounted for. The position may be derived from the location of the addressed object.

#### D.3.2 Theme Administrative Units

LandInfra specifies AdministrativeDivision, corresponding to INSPIRE Administrative Units. The AdministrativeDivision is a SpatialUnit with BoundingElements, corresponding to INSPIREs AdministrativeBoundary

The term 'condominium' is used in the context of INSPIRES Administrative Unit to denote 'independent administrative areas that are administered by two or more countries'. In LandInfra Condominium refers to a unit of real estate.

Administrative Units include maritime divisions. The scope of LandInfra is restricted to land, including inland water.

LandInfra provides the same associations as INSPIRE with respect to the administrative hierarchy, cf. the table below. The BoundingElements allow for complying with requirements 4 and 5 below.

INSPIRE	LandInfra
---------	-----------

Administrative Units	AdministrativeDivision
AdministrativeBoundary	BoundingElements of SpatialUnit
INSPIRE requirements	
1. upperLevelUnit association role of AdministrativeUnit	ownedBy
2. lowerLevelUnit association role of AdministrativeUnit	part
3. co-administered by two or more other administrative units	NCC
4. Administrative units at the same level of administrative hierarchy shall not conceptually share common areas	Applicable. Enclaves within another unit are accounted for
5. Instances of the spatial object type AdministrativeBoundary shall correspond to the edges in the topological structure of the complete (including all levels) boundary graph.	Accounted for
6. + 7. The spatial extent of a condominium ...	NCC

### D.3.3 Theme Buildings

In INSPIRE context, a building component is any sub-division or element of a building. More application schemas are applied, including 2D and 3D, and an Extended Base, with extended 2D and 3D. A Building may be composed of BuildingParts, but it is left to the data producer to define these parts. Yet, it is defined that 'A BuildingPart is a sub-division of a Building that might be considered itself as a building.' (5.3.2.1.4; 5.4.2.1.2; 5.5.2.1.2). The application schema BuildingsExtended2D specifies a BuildingUnit as a subdivision of Building with its own lockable access from the outside or from a common area (i.e. not from another BuildingUnit), which is atomic, functionally independent, and may be separately sold, rented out, inherited, etc. (5.7.2.2.1). The Buildings3Dmodel allows to provide the four levels of City GML. The extended 3D profile is available for providing more details.

INSPIRE		LandInfra
Building	somehow compares to	CondominiumBuilding
BuildingUnit	compares to condominiumMainPart of	CondominiumUnit
NCC		BuildingPart

#### D.3.4 Cadastral Parcels

In INSPIRE context, cadastral parcels should be forming a partition of national territory, and the same holds for LandInfra, which conceives a lotParcel as a part of the official tessellation of the territory of the jurisdiction (7.10.2.4). The LandInfra lotParcel is one among more types of LandParcel.

The basic object types/ classes, and next attributes, compares as follows:

INSPIRE spatial object types	LandInfra classes
Basic Property Unit, consisting of [1..*] CadastralParcel	PropertyUnit, consisting of [1..*] LandParcels, or of a CondominiumUnit
Cadastral Parcel	LandParcel
Cadastral Boundary	BoundaryElements of SpatialUnit
Cadastral Zoning	may be rendered through AdministrativeDistrict
INSPIRE attributes	LandInfra attributes
nationalCadastralReference	cadastralParcelID: identifier of a LandParcel, unique across an entire jurisdiction, or globally unique with the inclusion of an ID.scope value
Geometry Representation is restricted 0-, 1-, 2-dimensional geometries where all curve interpolations are linear.	BoundaryElements of SpatialUnit allow for a rich set of representations
The cadastral boundaries corresponding to	

the outline of a cadastral parcel shall form closed ring(s).	
<p>Temporality representation</p> <p>-beginLifespanVersion and end ... related to the life of the spatial object in the spatial data set (geographic/dataset point of view)</p> <p>-validFrom and validTo are related to the life of the entity in the real world (legal point of view).</p>	Datasets may be versioned internally, cf. 7.2.1.1.
areaValue: Area [0..1]	parcelArea: Area
referencePoint: GM_Point [0..1]	BoundingElement.BEpoint geometry:Point

The scope of LandInfra is infrastructure projects and the related surveying. This may explain the greater versatility in spatial representations, and the diminutive account for temporality.

#### D.3.5 Theme Coordinate Reference Systems (CRS)

INSPIRE	LandInfra
datum of the European Terrestrial Reference System 1989 (ETRS89) in areas within its geographical scope, or the datum of the International Terrestrial Reference System (ITRS) or other geodetic coordinate reference systems compliant with ITRS in areas that are outside the geographical scope of ETRS89.	
Three-dimensional Cartesian coordinates based on above datum and using the parameters of the Geodetic Reference System 1980 (GRS80) ellipsoid.	
Two-dimensional geodetic coordinates (latitude and longitude) based on above and using the parameters of the GRS80 ellipsoid. – Plane coordinates using either the ETRS89 Lambert Azimuthal Equal Area, the ETRS89 Lambert Conformal Conic, or the ETRS89 Transverse Mercator coordinate reference system.	
Compound CRS: a coordinate reference system that combines a two-dimensional CRS (the horizontal component) with a one-dimensional CRS (the vertical component).	

Coordinate reference system parameters and identifiers shall be managed in one or several common registers for coordinate reference systems, e.g. <a href="http://www.opengis.net/def/crs/EPSSG/0/">http://www.opengis.net/def/crs/EPSSG/0/</a> ...	
---	--

LandInfra supports the use of multiple CRS, even within a LandInfra dataset.

#### **D.3.6 Geographical Names**

In INSPIRE context, Geographical Names refer to names of areas, regions, localities, cities, suburbs, towns or settlements, or any geographical or topographical feature of public or historical interest. LandInfra provides for the optional use of geographical names for AdministrativeDistricts (adName, 7.10.3) and LandElements (elementName, 7.9.3). Spelling and multi-language use are not accounted for.

#### D. 4 The ePlan Model of the Intergovernmental Committee on Surveying and Mapping, Australia

The ePlan Protocol - LandXML Structural Requirements, (v1, released 12-11-2010, amended 15-12-2015) [17], is the most recent among ePlan documents. It supplements the ePlan Model (v 1.0 12-11-2010) and the ePlan Protocol - LandXML mapping (v 2.1. 6-08-2011). The two ePlan Protocols addresses

- LandXML structural requirements that specify how a Cadastral Information File (CIF) should be structured to correctly represent specific pieces of data
- Specification of every valid LandXML data element and attribute included in the ePlan CIF.

The ePlan Model regards 'conceptual level specifications that deal with the fundamental understanding of the ePlan model.'

LandInfra is a conceptual model. Therefore, it appears most appropriate to compare LandInfra with the ePlan model, cf. Nevil Cumerford (FIG, 2010). This is motivated also by the fact that the ePlan Model contains explanations which are outside the scope of the two Protocol documents.

<b>ePlan Model</b>	<b>LandInfra</b>
<b>2.1 DOCUMENT package</b>	
2.1.3 Dealing, i.e. the title information that is usually generated/ added to a plan upon lodgement. This is the internal key to the Titling system.	7.10.4 Statement +landRecordingDocumentID
2.1.5 Party, i.e. the person who is issuing the approval	Approval is out of scope, but see 7.10.4 Signatory and 7.2.1.8 Professional
2.1.1 Approval; 2.1.2 Condition; 2.1.4 Image	Out of scope
<b>2.2 SURVEY package</b>	
2.2.1 Document	7.2.1.3 Document, which inherits i.a. dateTime and datasetVersion
2.2.2 Survey, SurveyDesc, incl the description of the parcels being	7.10.4.1 StatementType with e.g. +cadastralAccount and

cancelled or the parcels being created.	+easementEstablishment
+PurposeOfSurvey, +SurveyType	7.8.1.1 Survey +purposeOfSurvey +surveyType
2.2.3 Survey Document, concerning administrative and textual data associated with the metadata of the survey, incl +surveyFormat,	7.10.2.3 LandParcel, 7.10.2.5 Off-surfaceUnit, and 7.11.1 CondominiumUnit account for the legal object surveyed.
+SchemeNo, +SchemaName,	CondominiumScheme, which inherits 7.2.1.3 DocumentID.
+SurveyFirm	7.2.1.8 Professional with +company and +registration (of license)
+SurveyorReference	7.10.4 Statement + caseID
2.2.4 Administrative Area, which may affect the processing of a plan	7.10.3 AdministrativeDivision provides for political, judicial and administrative divisions, to be mentioned e.g. in 7.10.4 Statement. Beach protection zones and similar may be referenced through 7.10.2.3 LandParcel + plannedLandUse
2.2.5 Survey Annotation	Would require further specification of 7.10.4 Statement
2.2.6 Coordinate System	7.2.1 LandInfra Dataset + crs
2.2.7 Surface	BoundingElement of the SpatialUnit of LandParcel
<b>2.3 PARCEL package</b> regarding the non-spatial elements of parcel.	
2.3.1 Parcel - A parcel can be made up of a number of sub parcels; each sub parcel must be a closed figure	7.10.2.2 LandPropertyUnit, comprise one or more 7.10.2.3 LandParcels which together constitute the LandPropertyUnit
+state, +parcelType	7.10.2.4 Land Parcel State +mainParcel. For

	Building Format lot, see below
+class	7.10.2 InterestInLand
+useOfParcel, +legalArea	7.10.2.3 +currentLandUse, +plannedLandUse, +parcelArea
2.3.1.2 Allocations (Linkages) is the process by which an existing interest is brought forward onto the parcel(s) being created, e.g. a survey of Lots 1-3 and Easement A in Lot 3 requests allocation between Easement A and Lot 3 be defined.	7.10.4.1 StatementType +cadastralAccount may include/ be supplemented with a Statement, accounting for each LandParcel the easements before and after the change.
2.3.2 Building Format Lot	7.11.1 CondominiumUnit with CondominiumScheme, CondominiumBuilding and BuildingPart provide i.a. for BuildingNumber and FloorNumber of 7.11.4 BuildingParts
2.3.3 Title References	Outside scope of LandInfra
2.3.4 Volumetric Lot	7.10.2.5 Off-surfaceUnit
2.3.5 Scheme Land	7.11.1 CondominiumUnit +shareInJointFacilities
2.3.6 Exclusion Area	To be addressed through 7.10.2.3 LandParcel and/ or 7.10.2.6 Easement
2.3.7 Plan Feature	7.10.2.6 Easement, the SpatialUnit of which may be 3D
<b>2.4 GEOMETRY package</b>	
2.4.1 Line, 2.4.2 RegularLine, 2.4.3 Irregular Line, 2.4.4 Curve, 2.4.5 Polygon, 2.4.6 Volume	7.10.7 BoundingElements
<b>2.5 POINTS package</b>	

2.5.1 Survey Points	7.8.7.1 SurveyResults
2.5.2 Survey Marks	7.2.1.4 Survey Mark, 7.10.5 SurveyMonument
<b>2.6 OBSERVATIONS package</b>	7.8 Survey
<b>2.7 SURVEYOR package</b>	
2.7.1 Surveyor	7.2.1.8 Professional
2.7.2 Surveyor Certificate	7.2.1.8 Professional +registration

## Annex E TIN Example

An example of a TIN is provided herein, along with a discussion on the expected behavior while constraints are added to the triangulation [18].

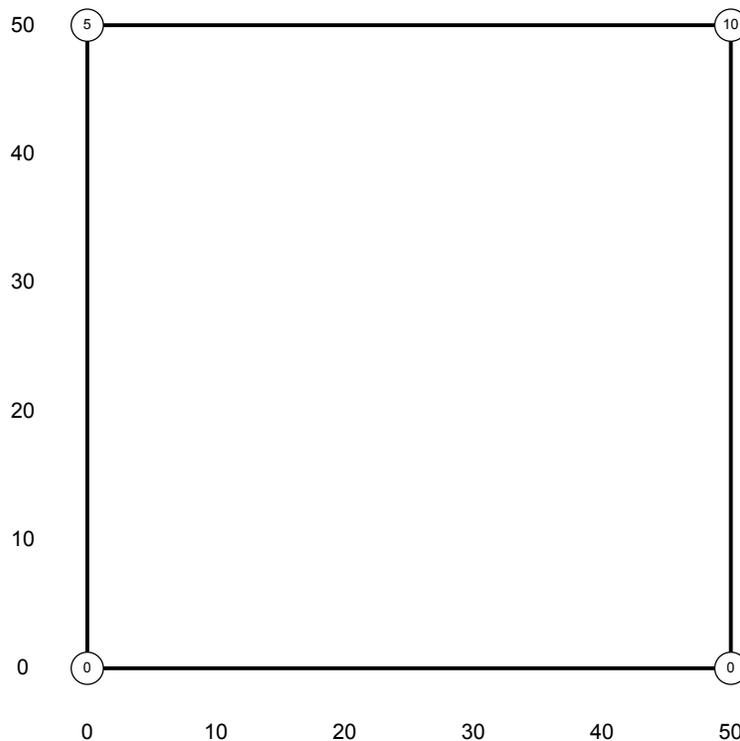
### E. 1 Input Points

A very simple surface serves as the basis for the example.

It is defined by four points.

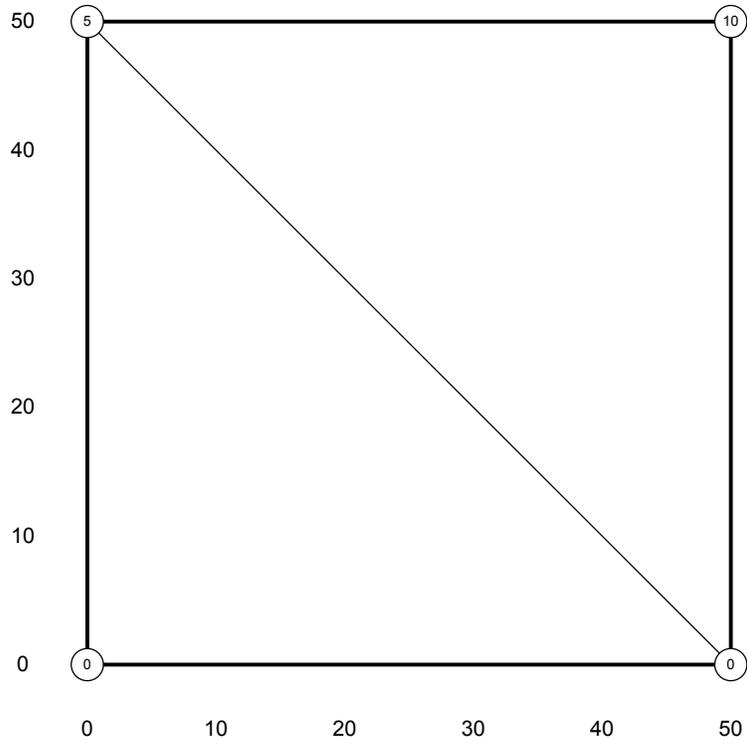
<b>X</b>	<b>Y</b>	<b>elevation</b>
0	0	0
0	50	5
50	0	0
50	50	10

This might be better visualized graphically. Point elevations appear inside the circles which represent the input points.



## E. 2 Initial Triangles

From just these four input points, a TIN containing two triangles can be constructed.

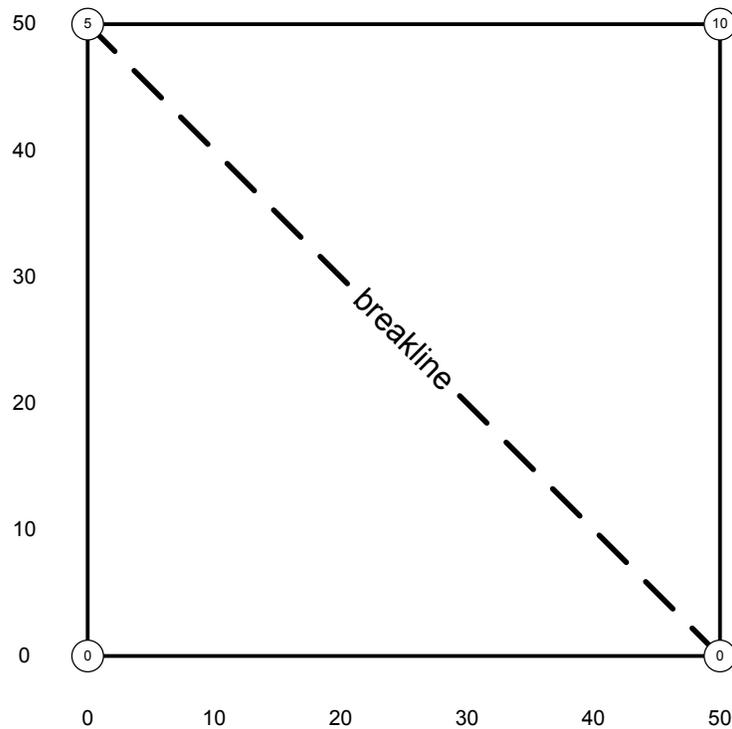


### E. 3 Constraints

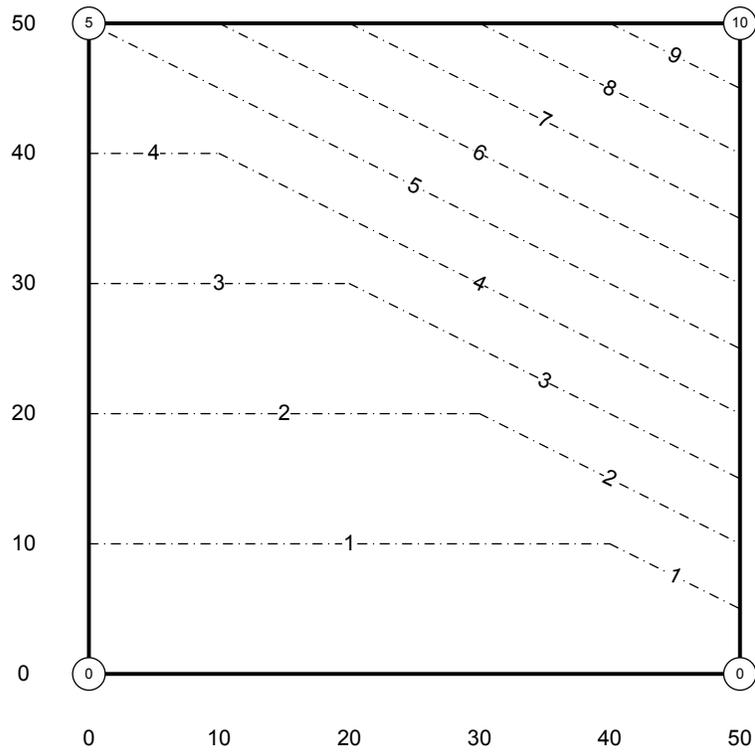
Several constraints can be added to the surface. These include: breaklines, soft breaks, control contours, break voids, drape voids, voids, holes, stop lines, maximum triangle side length and boundaries.

#### E.3.1 Breakline

A (hard) breakline represents a discontinuity in the slope of a TIN surface caused by a natural feature (local ridge or depression) or a man-made feature (road, building). To insure that the triangulation above breaks from  $(0,50)$  to  $(50,0)$ , instead of  $(0,0)$  to  $(50,50)$ , a breakline constraint is specified along the diagonal from  $(0,50,5)$  to  $(50,0,0)$ . When a breakline is specified, triangles must be adjusted so that no triangle is crossed by the breakline. Part or all of the breakline becomes an edge of two or more triangles. The elevation along the breakline takes precedence over the elevation of the original triangulated surface for the entire length of the breakline.

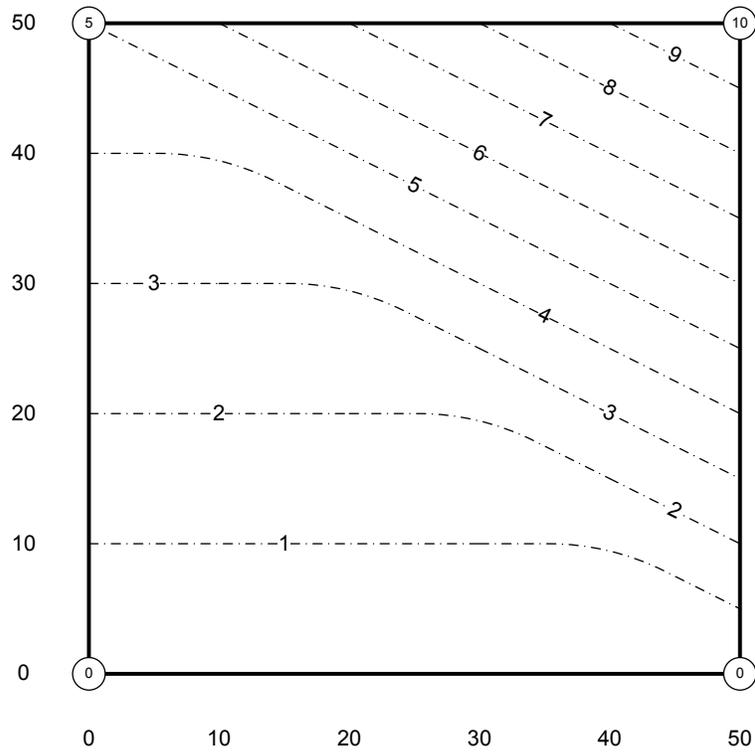


An output contour map of the resultant surface shows the effect of the breakline.



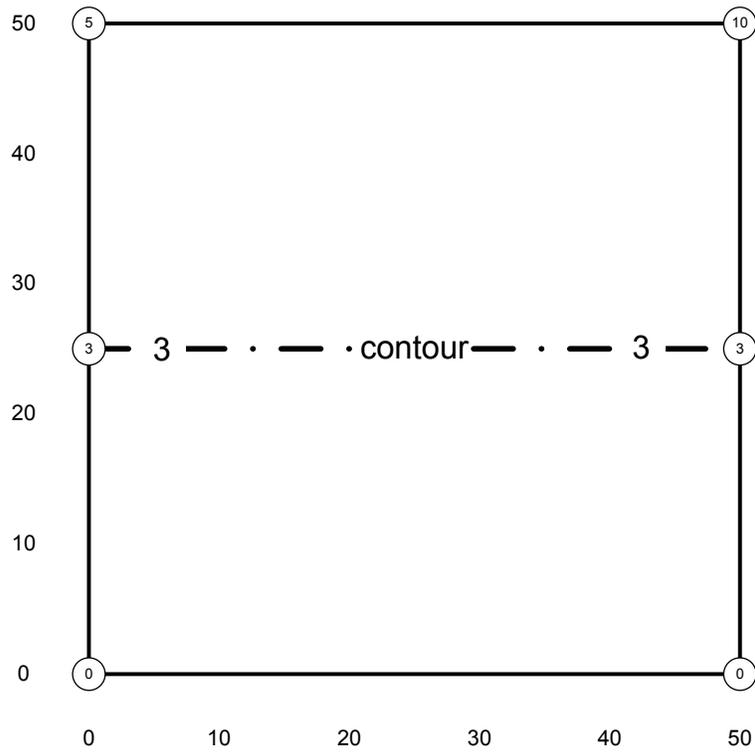
### E.3.2 Soft Break

A soft breakline is identical to a breakline except that the contours generated from the surface are smoothed.

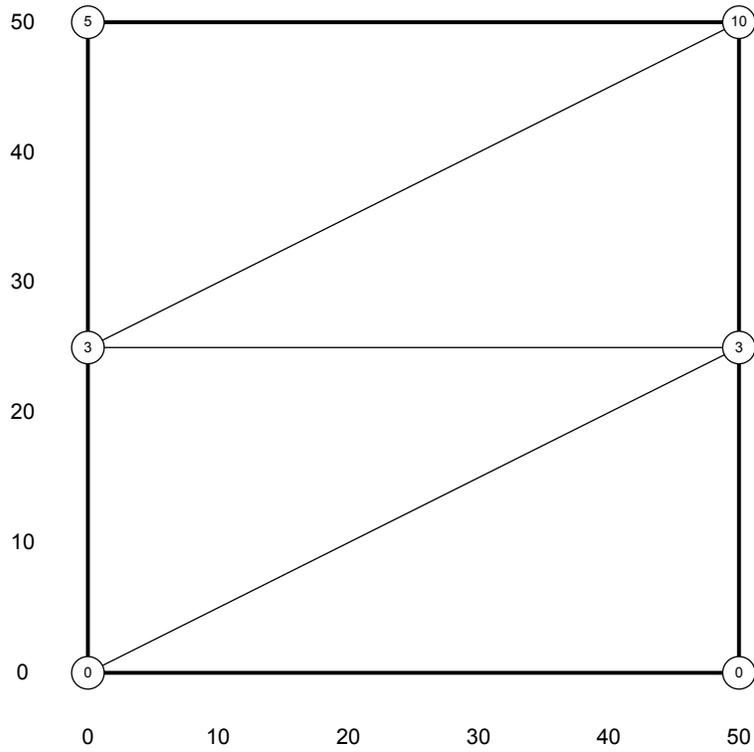


### E.3.3 Control Contour

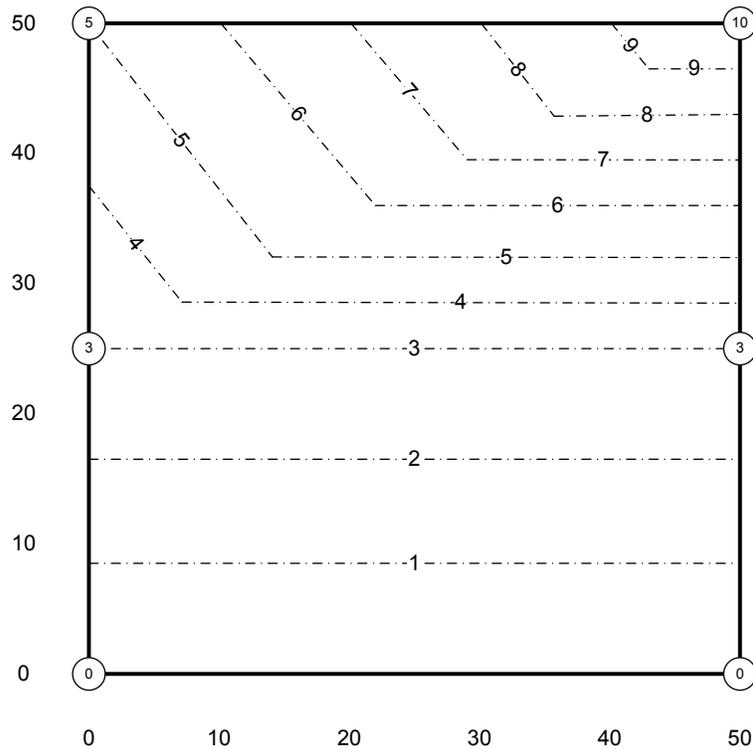
A control contour is a breakline having a constant elevation throughout its length. Instead of the breakline shown above, consider adding a control contour from (0,25) to (50,25) at an elevation of 3.



Introducing this control contour instead of the previous breakline would result in four triangles.



The output contour map of the surface (without smoothing) would look like:



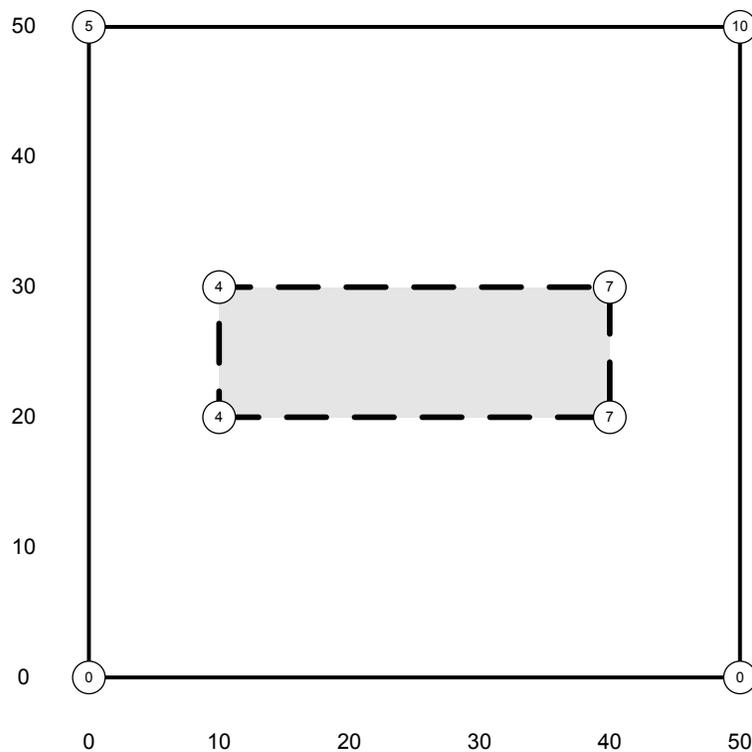
The reason to distinguish between breaklines and the more specific control contours is twofold. First, control contours are checked to verify that all vertices of the line are 3D and have the same elevation. Secondly, triangles within the vicinity of a control contour need to be assessed to insure they are not zero slope triangles (all three vertices fall on the same control contour).

### E.3.4 Break void

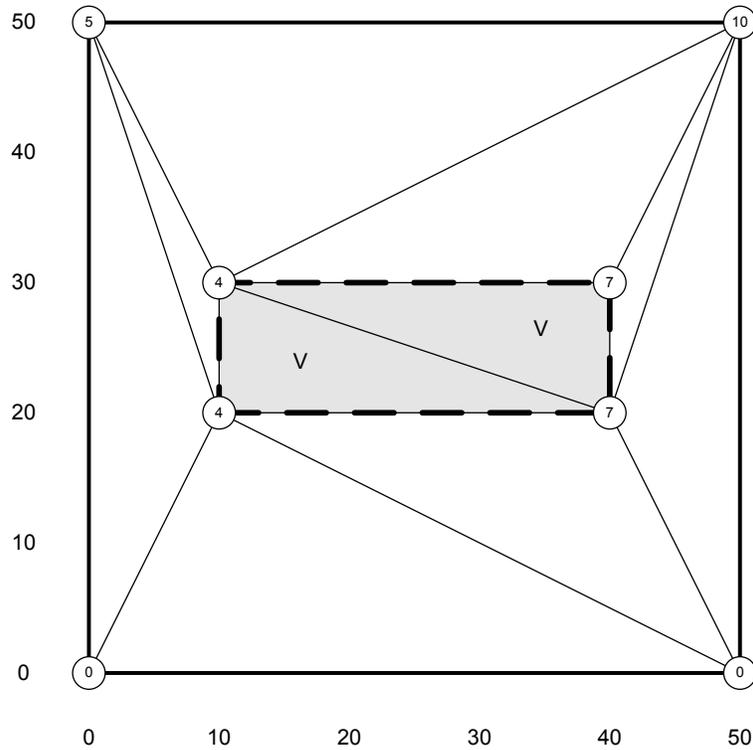
A break void is an area of a TIN surface, defined by a Polygon, which is to be voided. The triangles within this area still exist but are tagged as being void. For a break type void, the boundary of the void is treated as a breakline in that triangles must be adjusted so that no triangle is crossed by the break void boundary. Part or all of the break void boundary becomes an edge of two or more triangles. The elevation of this break void boundary takes precedence over the elevation of the original triangulated surface for the entire length of the boundary.

Given the triangulated surface in 0, a break void (shaded area in the figure below) is introduced having the following points as vertices of its exterior boundary:

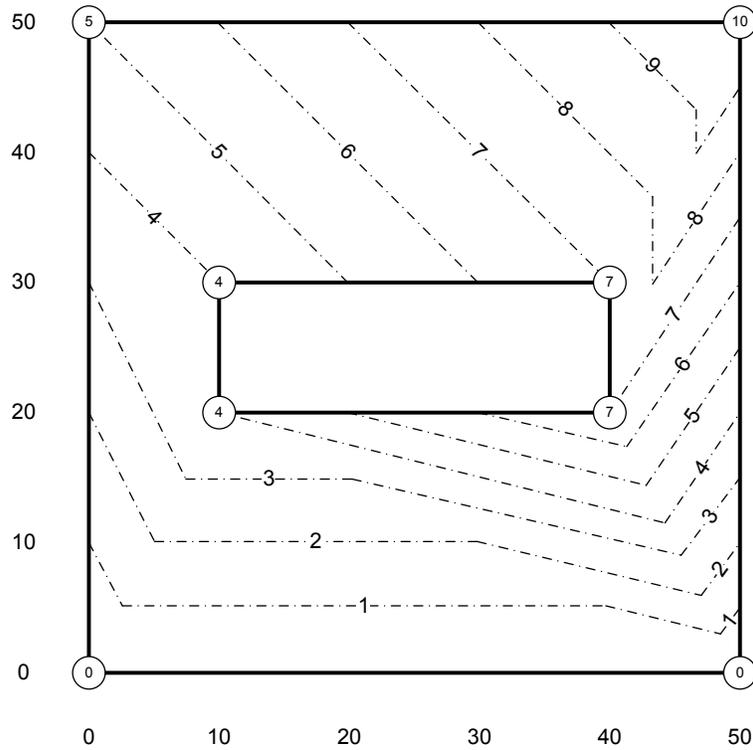
X	Y	Elevation
10	20	4
10	30	4
40	30	7
40	20	7



Because the two original triangles are crossed by the break void boundary, they must be adjusted. If possible, no additional vertices are introduced into the break void boundary. This results in the following triangulated surface. Triangles within the break void boundary are retained, but marked as void. "V" designates void triangles.



The output contour map of the surface with the break void added (without smoothing) would look like:



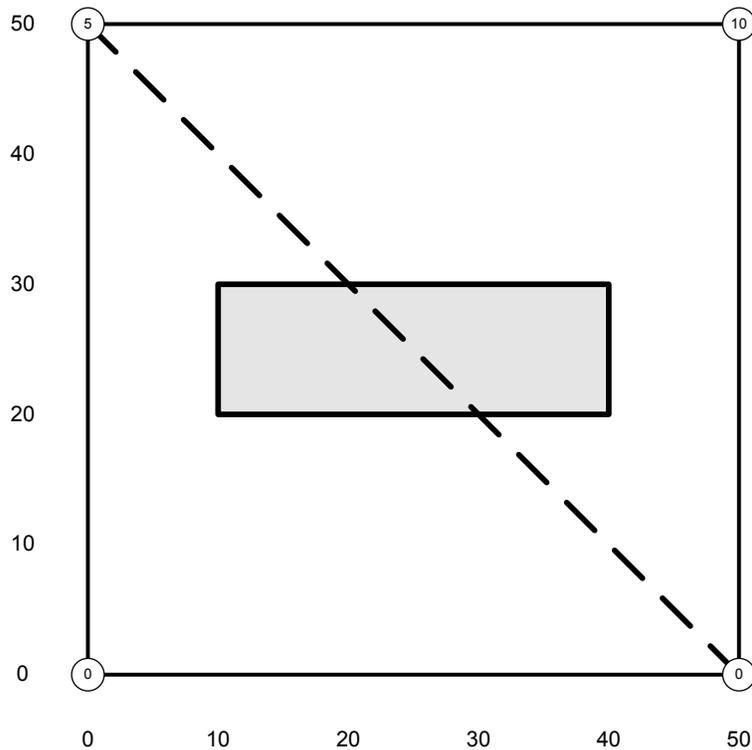
All void types have a geometry of type Polygon, so they may contain interior boundaries. These interior boundaries are treated in the same way as the exterior boundary. Triangles existing inside of the interior boundary are exterior to the Polygon and are therefore not in the void area. These triangles are therefore not marked as void.

### E.3.5 Drape void

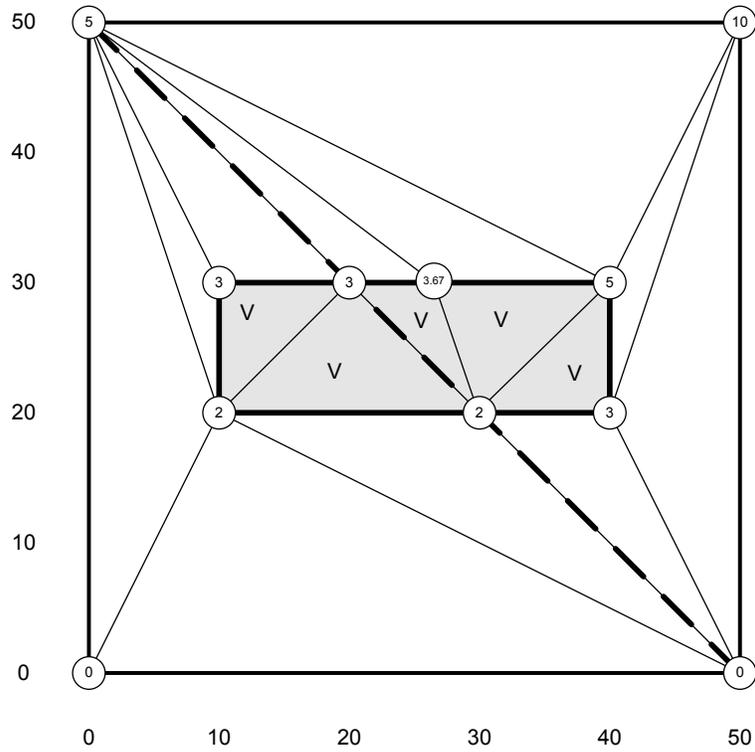
A drape void is an area of a TIN surface, defined by a Polygon, which is to be voided. The triangles within this area still exist but are tagged as being void. For a drape type void, the boundary of the void is like that of a break void in that triangles must be adjusted so that no triangle is crossed by the boundary. Part or all of the drape void boundary becomes an edge of two or more triangles. However, for drape voids, the elevation of the original triangulated surface takes precedence over the elevation of the drape void boundary. The geometry of a drape void is specified by a Polygon. If the Polygon is 3D, the elevation values are ignored.

Given the triangulated surface in 00 (including breakline but not the break void), a drape void (shaded area in the figure below) is introduced having the following points as vertices of its exterior boundary:

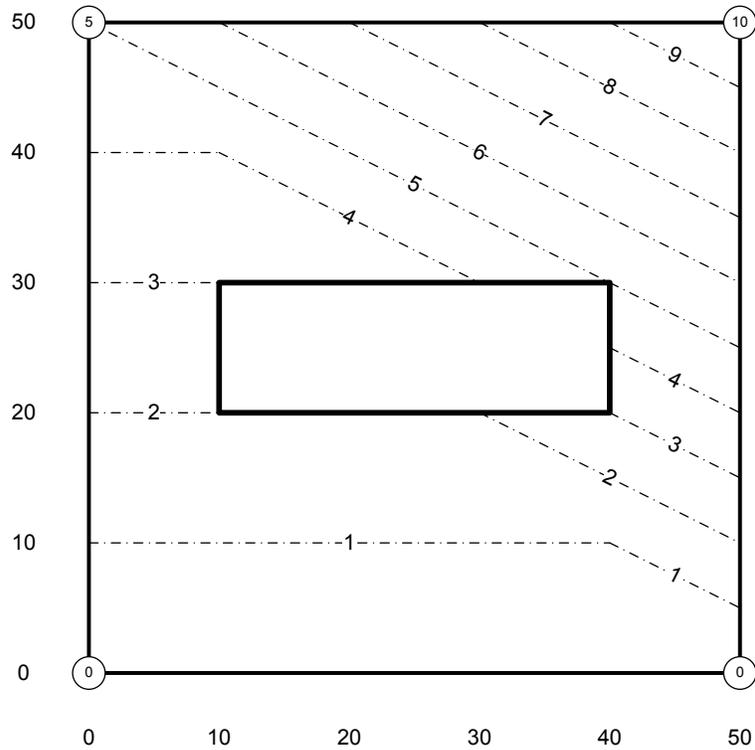
X	Y	Elevation
10	20	
10	30	
40	30	
40	20	



Because the two triangles are crossed by the drape void boundary, they must be adjusted. After drape void insertion and subsequent post processing to find and remove any redundant points that were added during the insertion, the result is the following triangulated surface:



As might be expected, the output contour map of the resultant surface with the breakline and drape void added, looks exactly like the one for the original surface with breakline since the drape void merely subtracts out a void area without affecting the surface elevations.

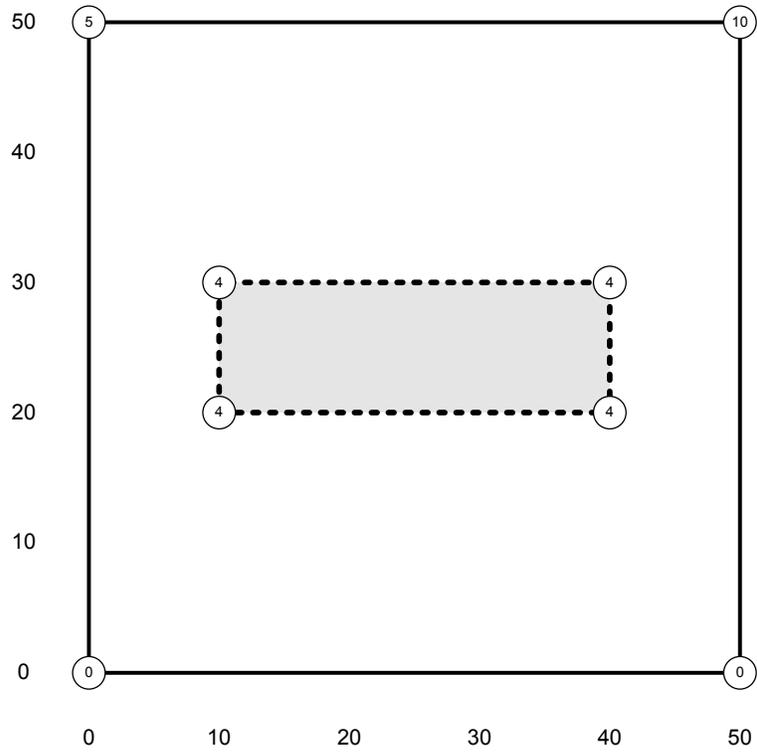


### E.3.6 Void

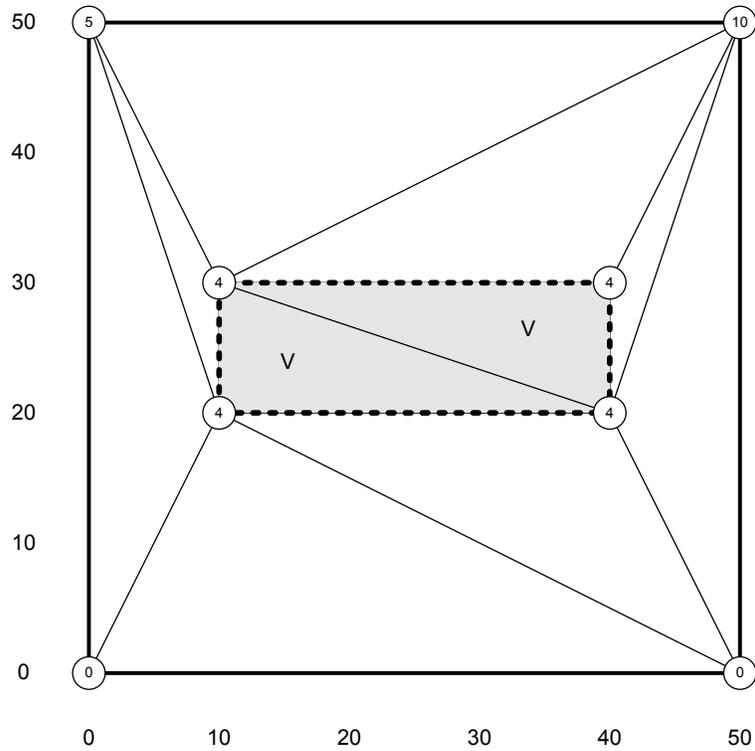
A regular void is an area of a TIN surface, defined by a Polygon, which is to be voided. The triangles within this area still exist but are tagged as being void. For a regular type void, the boundary of the void is like that of a break void in that triangles must be adjusted so that no triangle is crossed by the boundary. Part or all of the void boundary becomes an edge of two or more triangles. However, for regular voids, only the elevations of the void boundary vertices take precedence over the elevation of corresponding points on the original surface. That is, these vertices are treated as points for triangulating. The regular void boundaries between these vertices are handled as drape void boundaries - elevations from the original surface take precedence. The geometry of a regular void is specified by a Polygon.

Given the triangulated surface in 0, a regular void (shaded area in the figure below) is introduced having the following points as vertices of its exterior boundary:

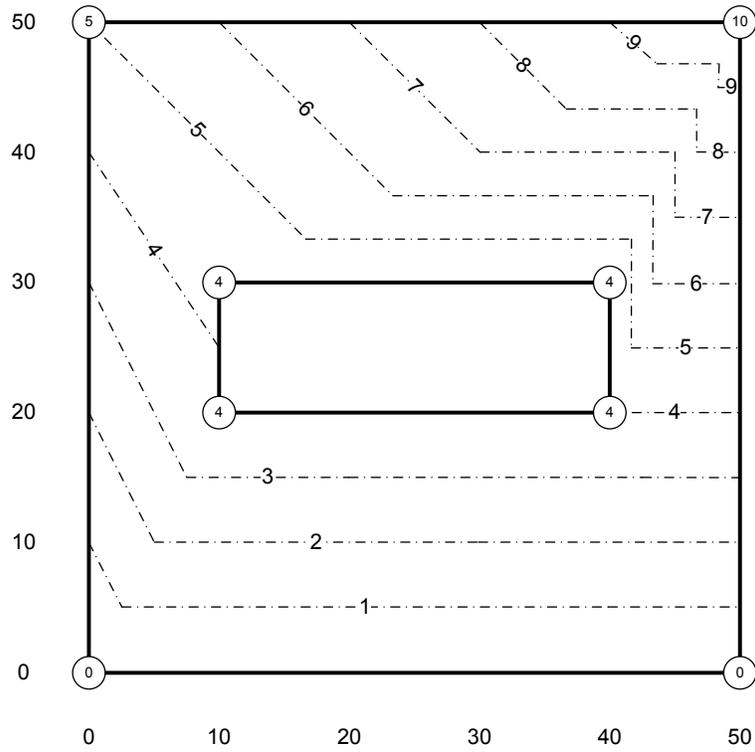
<b>X</b>	<b>Y</b>	<b>Elevation</b>
10	20	4
10	30	4
40	30	4
40	20	4



The four points representing the vertices of the regular void (including their elevations) are added to the initial four group spots and re-triangulation occurs. This results in the following triangulated surface:



The output contour map of the resultant surface with the regular void added would look like:



### **E.3.7 Hole**

A hole is an area of a TIN surface, defined by a Polygon, which is to be treated as a hole in the surface in that the triangles within this area still exist but are tagged as being a hole. The difference between a void and a hole is realized when two TIN surfaces are merged. When merging TIN surface A with TIN surface B, a void in surface A will take precedence over what is in the same area in B, the result being retention of the voided A triangles. A hole in surface A results in that part of surface B showing through the hole and becoming part of the resultant merged surface.

Hole boundaries are different than an interior boundary of a surface. The hole is still part of the interior of the surface since the triangles still exist. They are merely tagged as holes so applications know how to deal with them in a visibility context. This also allows for “islands” to exist inside the holes (if the hole polygon has interior boundaries). Here the triangles exist and are not tagged. Had we chosen to eliminate the triangles in the hole, islands would not be supported in a single surface.

Hole boundaries are like drape type void boundaries in that the elevation of the original triangulated surface takes precedence over the elevation of the hole boundary. The geometry of a hole is specified by a Polygon. If the Polygon is 3D, the elevation values are ignored.

### **Stop lines**

An alternative method of specifying void triangles with a drape void is to construct a linestring through the affected triangles. In order to insure that the stop line intersects the triangles, the elevation values of the stop line vertices are ignored if present.

Stop lines obviously only work for void areas involving a small number of triangles. Otherwise, it is easier to specify the void with a drape void polygon.

### **Maximum side length**

A maximum value may be defined for the length of a triangle side. Handling of this is implementation defined. Options are twofold. First, any triangle having a side whose length exceeds the limit is either deleted or is marked as void. Second, tested triangles may be any triangle in the surface or may be limited to those around the boundary of the surface.

### **Surface Boundary**

Because TIN is a subtype of Surface, it has an exterior boundary and zero or more interior boundaries.

The boundary is treated as breaklines in that triangles must be adjusted so that no triangle is crossed by a boundary. After triangles are adjusted along the boundary, any triangles falling outside of the exterior boundary or inside of an interior boundary are deleted from the surface.

As mentioned in E.3.7, this is different from holes. Islands of triangles can exist within an area of hole triangles but cannot exist within an interior boundary.

## Annex F Revision history (informative)

Date	Release	Author	Paragraph modified	Description
<b>20160107</b>	<b>1.0</b>	<b>Scarponcini, ed.</b>		<b>Initial release</b>

## Annex G Bibliography

- [1] LandXML.org, *LandXML-1.2*, July 29, 2008, downloaded from the LandXML.org web site on March 15, 2011.
- [2] Scarponcini, Paul, *InfraGML Proposal (13-121)*, OGC 13-121r1, November 5, 2013, [https://portal.opengeospatial.org/files/?artifact\\_id=56477](https://portal.opengeospatial.org/files/?artifact_id=56477).
- [3] Scarponcini, Paul, editor, *OGC 14-116 OGC Draft LandInfra Conceptual Model*, December 31, 2014.
- [4] buildingSMART International, *IFC Alignment*, <http://www.buildingsmart-tech.org/infrastructure/projects/alignment>, downloaded November 11, 2015.
- [5] Gröger, Gerhard, et al, editors, OGC 08-007r1 *OpenGIS® City Geography Markup Language (CityGML) Encoding Standard*, August 20, 2008.
- [6] Jiyeong Lee et al, *OGC 14-005r3 OGC® IndoorGML*, December 2, 2014.
- [7] ISO, *ISO 19152:2012 - Geographic information - Land Administration Domain Model (LADM)*, 2012.
- [8] Moon, Hyoun-Seok, et al, Extraction of Road Structure Elements for Developing IFC(Industry Foundation Classes) Model for Road, *Journal of the Korea Academia-Industrial cooperation Society*, Vol. 15, No. 2, pp. 1195-1203, 2014.
- [9] ISO, *ISO 19101:2014 - Geographic information — Reference model — Part 1: Fundamentals*, 2014.
- [10] OGC 08-131r3 *The Specification Model — A Standard for Modular specifications*, October 19, 2009.
- [11] Macdonald, Keith. "Professions." Blackwell Encyclopedia of Sociology. Ritzer, George (ed). Blackwell Publishing, 2007. Blackwell Reference Online. 16. October 2015
- [12] Des Ormeaux, Anne and Jean-Marie Lessard. Legal Dictionary of Property in Canada - Dictionnaire juridique de la propriété au Canada, [online], 2009. [dualjuridik.org, 6. November 2015]
- [13] Çağdaş, Volkan (2013) An Application Domain Extension to CityGML for immovable property taxation: A Turkish case study. *International Journal of Applied Earth Observation and Geoinformation*. 21 (1) 545-555. doi:10.1016/j.jag.2012.07.013

- [14] UN/ECE. Guidelines on Condominium Ownership of Housing for Countries in Transition. Geneva, June 2003
- [15] ISO, *ISO 16739:2013 - Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*, 2013.
- [16] INSPIRE Thematic Working Groups (2013-14) INSPIRE Data Specifications – Technical Guidelines (<http://inspire.ec.europa.eu/index.cfm/pageid/2>)
- [17] Intergovernmental Committee on Surveying and Mapping (2010) ePlan Model, v1.0. (<http://icsm.govspace.gov.au/files/2010/11/ICSM-ePlan-Model-v1.0.pdf>)
- [18] Scarponcini, Paul, *Triangles, Polyhedral Surfaces, and TINs*, INCITS H2-2008-192 and ISO/IEC JTC 1/SC 32/WG 4: FAO-014, September 22, 2008
- [19] UN/ECE (2004) Guidelines on Real Property Units and Identifiers, Geneva, Switzerland  
<http://www.unece.org/fileadmin/DAM/hlm/documents/Publications/guidelines.real.property.e.pdf>
- [20] European University Institute (2005) Real Property Law and Procedure in the European Union: General Report. Florence, Italy.  
<http://www.eui.eu/Documents/DepartmentsCentres/Law/ResearchTeaching/ResearchThemes/EuropeanPrivateLaw/RealPropertyProject/GeneralReport.pdf>
- [21] Hendrik Goossens: Dutch Civil Law. Right of superficies.  
<http://www.dutchcivillaw.com/content/dutchcivillaw022.htm> Accessed 19. April 2016