

# Open Geospatial Consortium

Submission Date: 2017-03-24

Approval Date: 2017-06-02

Publication Date: 2017-08-16

External identifier of this OGC® document: <http://www.opengis.net/doc/standard/infragml/part3/1.0>

Internal reference number of this OGC® document: 16-103r2

Version: 1.0

Category: OGC® Encoding Standard

Editor: Paul Scarponcini

## OGC InfraGML 1.0: Part 3 - Alignments - Encoding Standard

### Copyright notice

Copyright © 2017 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. 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:  
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.....	7
2. Conformance.....	7
3. References.....	8
4. Terms and Definitions.....	8
5. Conventions .....	8
5.1 Abbreviations .....	8
5.2 UML Package and Class Diagrams.....	9
5.3 Requirements.....	9
6. InfraGML Parts.....	10
7. Requirements Classes for this Part .....	12
7.1 Structural Overview of Requirements Classes.....	12
7.1.1 Requirement Classes Defined in This Part .....	13
7.1.2 Dependent Requirement Classes Defined in Other Parts .....	13
7.1.3 Other Standards upon which the Requirement Classes of this Part Depend	13
7.2 Requirements Class: Alignment.....	14
7.2.1 Implementation decisions regarding OGC 15-111r1 UML.....	14
7.2.2 Specific Requirements for this Requirements Class.....	16
8. Media Types for any data encoding(s).....	20
Annex A: Conformance Class Abstract Test Suite (Normative).....	21
A.1 Conformance class: Alignment .....	21
Annex B: Sample XML (Informative) .....	23
B.1 Complete Alignment XML .....	23
Annex C: Revision history .....	28
Annex D: Bibliography .....	29

## Figures

Figure 1. InfraGML Part Dependencies.....	11
Figure 2. LandInfra Requirements Classes grouped into InfraGML Parts.....	12
Figure 3. Requirements Classes for this Part and its Dependencies.....	13
Figure 4. Alignment Logical Model.....	16
Figure 5. Transition Curve UML.....	19

## **i. Abstract**

This OGC InfraGML Encoding Standard presents the implementation-dependent, GML encoding of concepts supporting land and civil engineering infrastructure facilities specified in the OGC Land and Infrastructure Conceptual Model Standard (LandInfra), OGC 15-111r1. Conceptual model subject areas include land features, facilities, projects, alignment, road, railway, survey (including equipment, observations, and survey results), land division, and condominiums.

InfraGML is published as a multi-part standard. This Part 3 addresses the Alignment Requirements Class from LandInfra.

## **ii. Keywords**

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

OGC document, LandInfra, InfraGML, infrastructure, civil, alignment

## **iii. Preface**

In order to achieve consensus on the concepts supporting land and civil engineering infrastructure facilities, a UML Conceptual Model, LandInfra, was approved as an OGC standard in August, 2016. This model provides a unifying basis for encodings including but not limited to InfraGML, including similar work in buildingSMART International. It can also provide a framework for discussing how other software standards relate to LandInfra.

As an OGC standard, LandInfra follows the OGC modular specification standard, OGC 08-131r3. Because of the breadth of LandInfra, its subject areas are divided into separate Requirements Classes. This InfraGML encoding similarly is divided into Requirements Classes which are then grouped into Parts. A Part may address multiple LandInfra Requirements Classes but each Requirements Class is addressed in a single part. Because Requirements Classes may depend on other Requirements Classes (see LandInfra Figure 1, “Requirements Classes as UML Packages with their dependencies”), the reader of this InfraGML Part may need to conform to Requirements Classes in other Parts as well.

Note that this InfraGML encoding standard is a target of LandInfra and therefore this standard conforms to the Requirements Classes in LandInfra. On the other hand, an application claiming conformance to this InfraGML encoding standard must conform to the Requirements Classes contained in this InfraGML standard.

There are several reasons for separating InfraGML into Parts. Because they are likely to have separate authors, the rate at which each Part is completed may vary. It would not be advisable to wait until all Parts complete before any can be released as separate OGC standards. Multiple Parts will also allow each subject to have its own standards life cycle. One Part can be updated independent of other Parts, subject to dependency constraints. And of course, it should be easier for the application software developer to only deal with Parts relevant to their application.

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

Swedish Transport Administration

Trimble Inc.

Autodesk

#### **v. Submitters**

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

<b>Name</b>	<b>Affiliation</b>
Paul Scarponcini, SWG chair	Bentley Systems, Inc.
Hans-Christoph Gruler, SWG co-chair	Leica Geosystems
Peter Axelsson	Swedish Transport Administration
Lars Wikström	Swedish Transport Administration
Leif Granholm	Trimble Inc.
Johnny Jensen	Trimble Inc.
Thomas Liebich	buildingSMART International
Orest Halustchak	Autodesk

## 1. Scope

InfraGML is a GML encoding standard of the LandInfra Conceptual Model standard, OGC 15-111r1. InfraGML is provided as a set of individual though inter-dependent Parts, each of which is a GML standard.

The overall scope of this InfraGML Encoding Standard is infrastructure facilities and the land on which they are constructed. Also included is the surveying necessary for the setting out and as-built recording of these facilities and land interests. Primarily having a civil engineering point of view, InfraGML is relevant across all life cycle phases of a facility. Subject areas include land features, facilities, projects, alignment, road, railway, survey (including equipment, observations, and survey results), land division, and condominiums.

The scope of this Part 3 of InfraGML addresses the following subject area: alignment. The InfraGML Alignment Requirements Class is included. It is optional in that an application can conform to InfraGML without supporting it, for example by only supporting the Survey Requirements Classes in Part 6. However, to claim support for Alignment, an application must also support the InfraGML Core Requirements Class.

## 2. Conformance

The InfraGML encoding standard defines requirements, grouped into Requirements Classes, for applications which read and write information about infrastructure facilities and the land on which they are constructed, including the surveying necessary for the setting out and as-built recording of these facilities and land interests.

The OGC modular specification (OGC 08-131r3) defines “standardization target” as the entity to which requirements of a standard apply. It further notes that the standardization target is the entity which may receive a certificate of conformance for a requirements class. The standardization target type for this standard is therefore:

- software applications which read/write data instances, i.e. XML documents that encode land, infrastructure facility, and survey data for exchange

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 encoding standard, a standardization target shall choose to implement the core conformance class and any of the other conformance classes with

---

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

their dependencies. Conformance classes are based on Requirements Classes which are specified in this and possibly other Parts of the InfraGML standard.

All requirements classes and conformance classes described in this document are owned by the standard(s) identified. Note that Conformance Classes for this Part of InfraGML may require conformance with Conformance Classes from other Parts of InfraGML.

### **3. References**

The following normative documents contain provisions that, through reference in this text, constitute provisions of this Part of InfraGML. 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.

OGC: OGC 07-036, *OpenGIS® Geography Markup Language (GML) Encoding Standard*, v3.2.1, 2007

OGC: OGC 10-129r1, *OGC® Geography Markup Language (GML) — Extended schemas and encoding rules*, v3.3, 2012

OGC: OGC 15-111r1, *OGC Land and Infrastructure Conceptual Model Standard (LandInfra)*, v1.0, 2016.

OGC: OGC 16-100, *OGC InfraGML 1.0: Part 0 – LandInfra Core – Encoding Standard*, v1.0, 2017

### **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.

The LandInfra standard contains a long list of terms and definitions relevant to the scope of InfraGML. As these will not be repeated here, the reader is directed to Clause 4 of LandInfra.

### **5. Conventions**

#### **5.1 Abbreviations**

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

bSI            buildingSMART International



GML	Geography Markup Language
ISO	International Organization for Standardization
OGC	Open Geospatial Consortium
UML	Unified Modeling Language
XML	eXtensible Markup Language

## 5.2 UML Package and Class Diagrams

The LandInfra standard contains UML diagrams for the concepts supported by InfraGML. As these will not be repeated here, the reader is directed to Clause 7 of LandInfra. UML will only appear in InfraGML in the rare cases where LandInfra is extended by InfraGML.

## 5.3 Requirements

When referred to in a Requirement or Requirements Class, the boxes contained in the LandInfra UML figures may all be called “Classes” even if they are data types, enumerations, code lists, unions etc. In most cases, these will be encoded as XML elements in InfraGML.

When an InfraGML Requirement states that “A conforming application shall support the [Requirements Class] XML elements listed in Table <n> in accordance with the GML XSD in this standard.”, the XSD was developed to support the UML for the corresponding LandInfra Requirements Class as follows:

- a) all classes shown as blue boxes for the corresponding LandInfra Requirements Class UML diagrams;
- b) all attributes, attribute cardinalities, and attribute data types of these classes (usually shown in subsequent diagrams);
- c) all associations, navigation, roles, and role cardinalities connected to the blue classes;
- d) all classes shown as beige boxes (another Requirements Class) in the diagrams connected to the blue box classes by association or used as attribute data types; and
- 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.

Note that, in rare cases, the OGC 15-111r1 UML may be altered. In such cases, the alterations are declared in the first subclause of each Requirements Class, entitled “Implementation decisions regarding OGC 15-111r1 UML”. Logical Model UML diagrams may be included if the implementation constraints of GML (or XML) dictate that the Conceptual Model cannot be implemented directly as shown in OGC 15-111r1.

In most cases, the InfraGML XML derived from the LandInfra UML follows the rules in OGC 07-036, GML, Annex E, UML-to-GML application schema encoding rules.

The only normative version of the GML XSD (XML schema definition) for all Parts of the InfraGML Encoding Standard is available from the official OGC XML schema repository at <http://schemas.opengis.net>. Any occurrences of all or part of this XSD contained within this document are to be considered to be informative only.

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

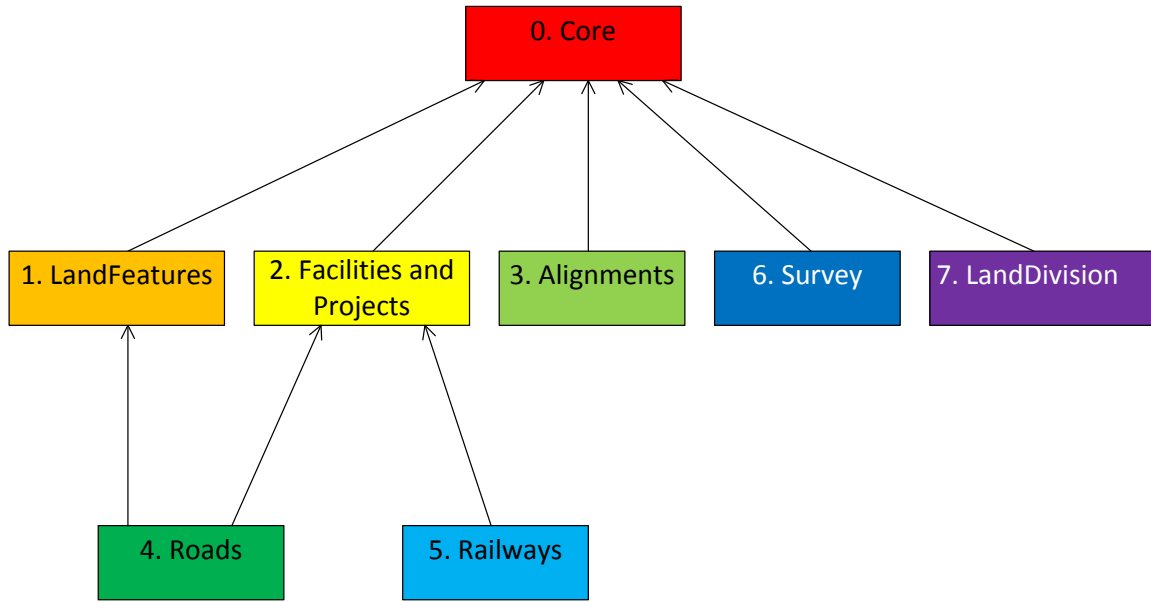
The URI base for this InfraGML encoding standard is <http://www.opengis.net/spec/infragml/part3/1.0>. All URIs of Requirements Classes, Requirements, and Conformance Classes contained in this standard are relative to this base.

## 6. InfraGML Parts

The InfraGML encoding standard has been divided into Parts. These Parts enable the grouping of LandInfra subject areas (Requirements Classes) into individual OGC encoding standards. All of these InfraGML encoding standards have a similar name: “OGC 16-10n, OGC® InfraGML 1.0: Part n - <part name> Encoding Standard”, where Part numbers and names are as follows:

n	<part name>
0	LandInfra Core
1	LandInfra LandFeatures
2	LandInfra Facilities and Projects
3	LandInfra Alignments
4	LandInfra Roads
5	LandInfra Railways
6	LandInfra Survey
7	LandInfra LandDivision

Some InfraGML Parts depend upon other parts:



*Figure 1. InfraGML Part Dependencies*

The boxes above represent InfraGML Parts. Arrows show Part dependencies.

The Part dependencies derive from the dependencies of the InfraGML Requirements Classes contained in these Parts. The reader should rely more on the InfraGML Requirements Class dependencies and only use the Part dependencies as a guide for knowing which InfraGML Part standards to consider.

InfraGML Parts include the following LandInfra 1.0 Requirements Classes (UML Packages):

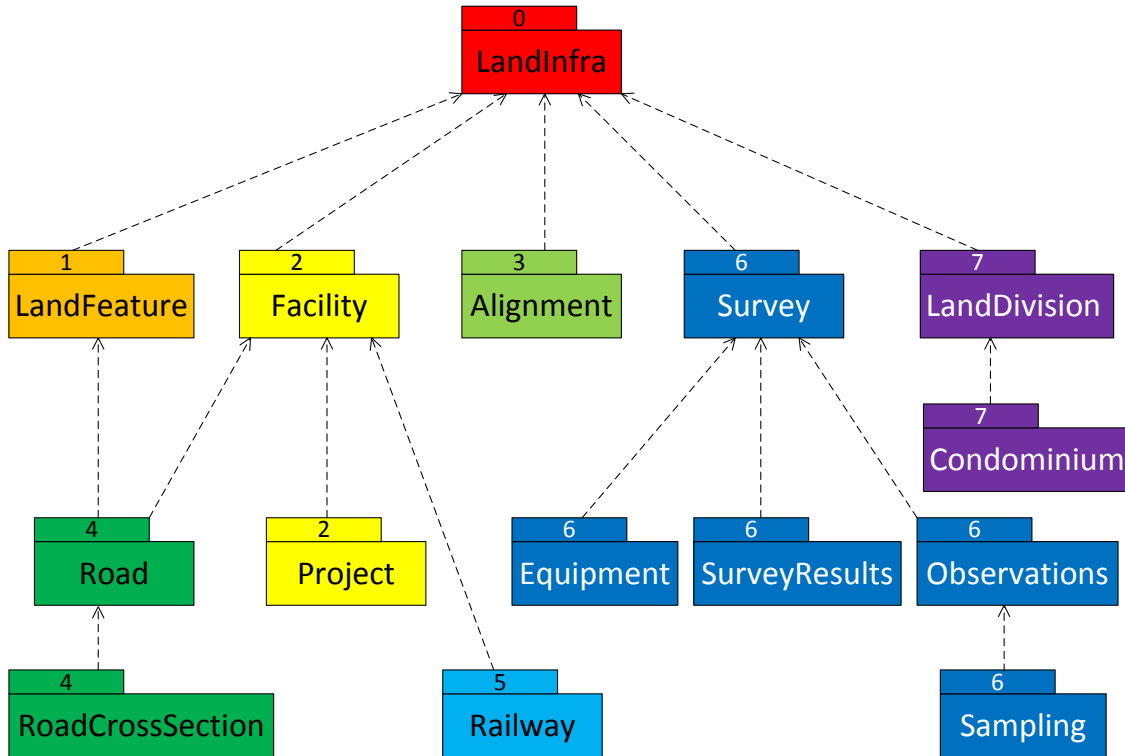


Figure 2. LandInfra Requirements Classes grouped into InfraGML Parts

The boxes above and their names represent LandInfra Requirements Classes. The numbers are InfraGML Part numbers. Dependency arrows shown above are dependencies between LandInfra Requirements Classes.

## 7. Requirements Classes for this Part

### 7.1 Structural Overview of Requirements Classes

The Requirements Classes for this Part of the InfraGML encoding standard (shown in blue in Figure 3 below) are defined in this Clause 7. Requirements Classes from other Parts upon which this Part's Requirements Classes are dependent (shown in beige in Figure 3 below) are listed here but defined in the documentation of their respective Parts. External OGC and ISO standards on which Requirements Classes in this Standard depend (shown in pink in Figure 3 below) are also listed. Below is a brief summary of the function of each of these Requirements Classes.

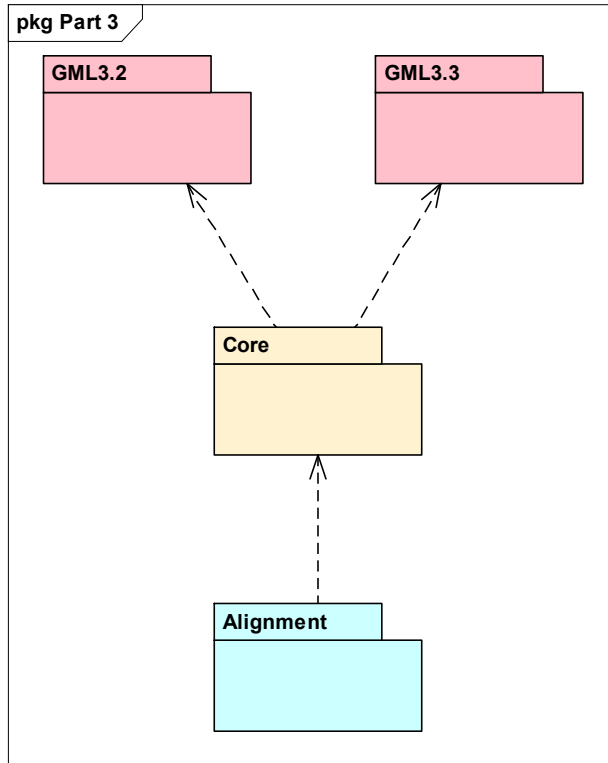


Figure 3. Requirements Classes for this Part and its Dependencies

### 7.1.1 Requirement Classes Defined in This Part

#### Alignment

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

### 7.1.2 Dependent Requirement Classes Defined in Other Parts

The Requirements Classes defined in this Part are dependent on the following Requirements Classes from other Parts.

#### Part 0 LandInfra Core

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.

### 7.1.3 Other Standards upon which the Requirement Classes of this Part Depend

For external OGC and ISO standards on which Requirements Classes in this Standard depend, a brief summary of the function of each of these Standards is described below.

## GML 3.2

OGC 07-036, OpenGIS® Geography Markup Language (GML) Encoding Standard, v3.2 provides most of the geometry types (e.g., Point, LineString, Polygon) used for spatial representations in this Standard. Defines Coordinate Reference Systems. Supports the General Feature Model upon which this Standard is based.

## GML 3.3

OGC 10-129r1, OGC® Geography Markup Language (GML) — Extended schemas and encoding rules, v3.3 defines the linear referencing concepts (e.g., linear element, distance along, Linear Referencing Methods) used for linearly referenced locations in this Standard.

### 7.2 Requirements Class: Alignment

Requirements Class	
<b>/req/alignment</b>	
Target type	Conforming application
Name	Alignment
Dependency	<b>/req/core</b> (from InfraGML Part 0)
Requirement	<b>/req/alignment/elements</b>
Requirement	<b>/req/alignment/description</b>
Requirement	<b>/req/alignment/CRS</b>
Requirement	<b>/req/alignment/measures</b>
Requirement	<b>/req/alignment/transitions</b>

#### 7.2.1 Implementation decisions regarding OGC 15-111r1 UML

The following implementation decisions have been made regarding the OGC 15-111r1 Alignment Requirements Class UML:

1. XML requires attribute names to begin with a letter, so Alignment.3DAlignment is changed to Alignment.alignment3D, Alignment.2DHorizontalType.2DLinestringRepresentation to Alignment.2DHorizontalType.linestring2DRepresentation, and Alignment.3DType.3DLinestringRepresentation to Alignment.3DType.linestring3DRepresentation.
2. Because Element was dropped in Part 2, elementID was lost for all Element subtypes, of which Alignment is one (via PositioningElement). Therefore, Alignment.alignmentID has been added.
3. Alignment cannot inherit from both PositionExpression (to allow it to be in a LandInfraDataset) and LinearElement (to allow for linearly referenced locations along it) since GML does not support multiple inheritance. In order for Alignment to behave as a LinearElement, then, it needs to support the Topic 19 ILinearElement interface. For

GML3.3, this was achieved by adding defaultLRM, measure, and startValue to gmlr:LinearElement. Therefore:

- a. li:LinearElement is created as a subtype of gmlr:LinearElement and extended to include a Referent property, since Linear Elements own Referents;
- b. PositioningElement is deleted as, with no properties of its own, it serves no useful purpose at this time. The Alignment RC is therefore directly dependent on Core, not on Facility where PositioningElement previously resided. All other non-Core RCs will therefore need to support the Alignment RC if they wish to include Alignments or AlignmentCurves.
- c. Alignment is then made a direct subtype of Feature so it can behave as a gmlr: or li:LinearElement.

4. For consistency with bSI, purpose and designAlternative have been added to Alignment as optional, CharacterString attributes.

5. A recent need has been expressed to be able to use the geometry as defined for Alignments for other purposes, such as Road StringLines. Therefore, Alignment.horizontal and Alignment.vertical have been moved to a newly introduced Curve subtype called AlignmentCurve. AlignmentCurve is then added back into Alignment as Alignment.geometry.

Alignment2DHorizontal is also made a Curve subtype. This enables two Alignments to be defined with the same horizontal but different verticals as was originally prescribed by LandInfra.

Alignment.alignment3D is moved to Alignment.

bSI is considering a similar move. It does not change any concepts but does make this way of defining an alignment curve geometry more widely available.

6. Transition Curves are needed for Railway Alignments but are shown as a future type called Spiral in OGC 15-115.r1. They are not in GML 3.2. They have therefore been renamed Transition Curves and added in this Alignment Requirements Class (see 7.2.2.2).

The net result of these decisions is captured in the Alignment Logical Model shown in Figure 4.

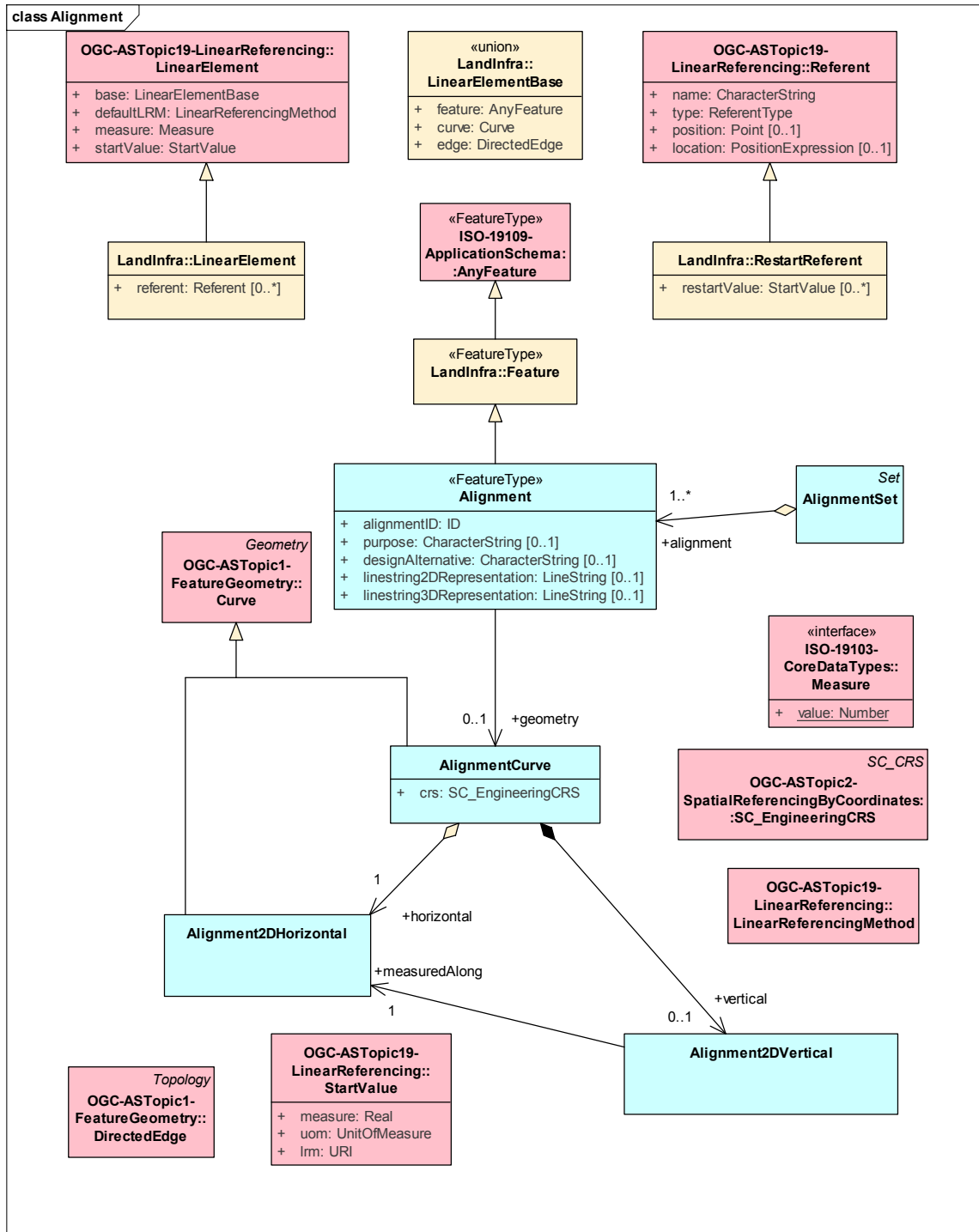


Figure 4. Alignment Logical Model

## 7.2.2 Specific Requirements for this Requirements Class



<b>Requirement</b>	<b><a href="#">/req/alignment/elements</a></b>  A conforming application shall support the Alignment XML elements listed in Table 1 in accordance with the GML XSD specified in <a href="http://schemas.opengis.net/infragml/part3/1.0/alignment.xsd">http://schemas.opengis.net/infragml/part3/1.0/alignment.xsd</a> .
--------------------	---

An application conforming to this standard shall support the Alignment XML elements listed below in Table 1 in accordance with the GML XSD specified in <http://schemas.opengis.net/infragml/part3/1.0/alignment.xsd>. Alignment XML element names are shown with a XML namespace prefix of “lia”. Corresponding LandInfra UML classes are shown with their LandInfra Requirements Class prefix of “Alignment”.

<b>InfraGML XML element</b>	<b>LandInfra UML Class</b>
lia:Alignment	Alignment::Alignment
lia:Alignment2DHorSegment	Alignment::Alignment2DHorSegment
lia:Alignment2DHorizontal	Alignment::Alignment2DHorizontal
lia:Alignment2DVertSegment	Alignment::Alignment2DVertSegment
lia:Alignment2DVertical	Alignment::Alignment2DVertical
lia:AlignmentCurve	InfraGML Alignment::AlignmentCurve
lia:AlignmentSet	Alignment::AlignmentSet
lia:TransitionCurve	InfraGML Alignment::TransitionCurve

*Table 1. InfraGML Alignment XML elements with corresponding LandInfra UML classes*

### 7.2.2.1 Additional Requirements

The description, CRS, and measures Requirements are from OGC 15-111r1. For further description of these, see that document.

<b>Requirement</b>	<b><a href="#">/req/alignment/description</a></b>  A conforming application 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.
--------------------	--

<b>Requirement</b>	<b><a href="#">/req/alignment/CRS</a></b>  The CRS of the geometry of an AlignmentCurve shall be an Engineering coordinate reference system.
--------------------	--

<b>Requirement</b>	<p><b><a href="#">/req/alignment/measure</a></b></p> <p>If an Alignment is used as a linear element, then the distanceAlong shall be measured along the Alignment Feature, the offsetLateralDistance shall be measured normal to the Alignment Feature, and offsetVerticalDistance shall be measured vertically up or down from the alignment Feature.</p> <p>If the LineString Curve geometry of an Alignment.linestring2DRepresentation or of an Alignment.Linestring3DRepresentation is specified as a linear element, then distanceAlong shall be measured along the LineString geometry and offsetLateralDistance and offsetVerticalDistance values shall be measured normal to the LineString geometry from this DistanceAlong point.</p> <p>If an Alignment.geometry AlignmentCurve 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 Curve. OffsetVerticalDistance values shall be vertical and 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>
--------------------	--

### 7.2.2.2 Transition Curve

OGC Abstract Specification Topic 1, Geometry, currently supports LineString, CircularString, and Clothoid types for use as CurveSegment2DHorizontal segment types. Alignments for Railways may also require Transition Curves of various types. UML for Transition Curves is included below as Figure 5.

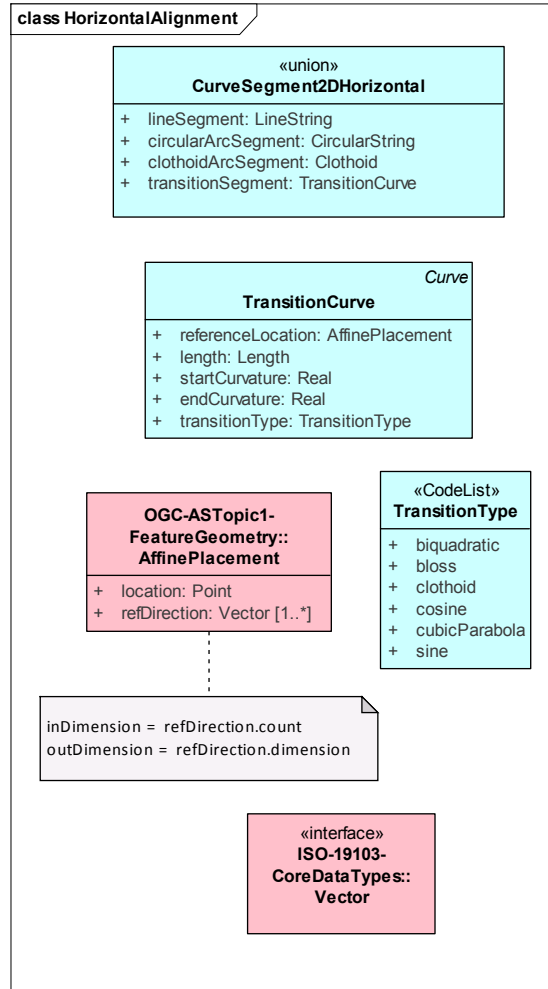


Figure 5. Transition Curve UML

TransitionCurve has the following attributes:

**referenceLocation**: an affine mapping of type gml:AffinePlacementType that places the spiral into the coordinate reference system of this curve. The spiral start point is the origin in the placement coordinates, and the initial direction is along the positive x axis.

**length**: the length of the TransitionCurve

**start curvature**: the start curvature value

**end curvature**: the end curvature value

**transitionType**: the type of TransitionCurve, selected from the TransitionType CodeList.

Curvature equals  $1/r$ , where  $r$  is the radius of curvature.

A conforming application shall specify which types of TransitionCurve it supports and, for each supported type, the application shall provide the curvature function it uses for that type.

<b>Requirement</b>	<b><a href="#">/req/alignment/transitions</a></b>  A conforming application shall specify which types of TransitionCurve it supports and, for each supported type, the application shall provide the curvature function it uses for that type.
--------------------	--

## 8. Media Types for any data encoding(s)

Data for all Parts of the InfraGML encoding standard is encoded in GML-conformant XML documents. The standard MIME-type and sub-type for GML data should be used to indicate the encoding in internet exchange, as specified in MIME Media Types for GML, namely ‘application/gml+xml’.

## Annex A: Conformance Class Abstract Test Suite (Normative)

### A.1 Conformance class: Alignment

<b>/conf/alignment</b>		
Requirements	<a href="#">/req/alignment</a>	
Dependency	<a href="#">/conf/core</a> (from InfraGML Part 0)	
Test	<b>/conf/alignment/elements</b>	
	Requirement	<a href="#">/req/alignment/elements</a>
	Test purpose	Verify that the conforming application supports the Alignment XML elements listed in Table 1 in accordance with the GML XSD specified in <a href="http://schemas.opengis.net/infragml/part3/1.0/alignment.xsd">http://schemas.opengis.net/infragml/part3/1.0/alignment.xsd</a> .
	Test method	Inspect the GML output 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 the conforming application specifies 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 application to verify the above requirement is satisfied.
	Test type	Capability
Test	<b>/conf/alignment/CRS</b>	
	Requirement	<a href="#">/req/alignment/CRS</a>
	Test purpose	Verify that the CRS of the geometry of an AlignmentCurve shall be an Engineering coordinate reference system.
	Test method	Inspect the GML output to verify the above requirement.
	Test type	Capability
Test	<b>/conf/alignment/measures</b>	
	Requirement	<a href="#">/req/alignment/measures</a>

	Test purpose	<p>Verify that the conforming application specifies that:</p> <p>If an Alignment is used as a linear element, then the distanceAlong shall be measured along the Alignment Feature, the offsetLateralDistance shall be measured normal to the Alignment Feature, and offsetVerticalDistance shall be measured vertically up or down from the alignment Feature.</p> <p>If the LineString Curve geometry of an Alignment.linestring2DRepresentation or of an Alignment.Linestring3DRepresentation is specified as a linear element, then distanceAlong shall be measured along the LineString geometry and offsetLateralDistance and offsetVerticalDistance values shall be measured normal to the LineString geometry from this DistanceAlong point.</p> <p>If an Alignment.geometry AlignmentCurve 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 Curve. OffsetVerticalDistance values shall be vertical and 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 application to verify the above requirement is satisfied.
	Test type	Capability
Test		<b><a href="#">/conf/alignment/transitions</a></b>
	Requirement	<b><a href="#">/req/alignment/transitions</a></b>
	Test purpose	Verify that the conforming application specifies which types of TransitionCurve it supports and that, for each supported type, the application provides the curvature function it uses for that type.
	Test method	Inspect the application to verify the above requirement is satisfied.
	Test type	Capability

## Annex B: Sample XML (Informative)

The following XML instance document attempts to demonstrate the use of most all of the elements supported by the specified Requirements Class(es), including all optional properties. All values are exemplary only and not intended to represent actual real world instance values. Not all xlink references are resolvable within this document.

### B.1 Complete Alignment XML

Example from Part3Alignment0410.xsd

Note: A value of **1** is merely a placeholder for a double precision number.

```
<?xml version="1.0" encoding="UTF-8"?>
<LandInfraDataset
  xmlns="http://www.opengis.net/infragml/core/1.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  gml:id="ds1"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:li="http://www.opengis.net/infragml/core/1.0"
  xmlns:lia="http://www.opengis.net/infragml/alignment/1.0"
  xmlns:gmlr="http://www.opengis.net/gml/3.3/lr"
  xmlns:gmlro="http://www.opengis.net/gml/3.3/lro"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/infragml/alignment/1.0 Part3Alignment0410.xsd">
  <datasetID>
    <ID>
      <identifier>DS3</identifier>
      <scope>OGC LandInfraSWG</scope>
    </ID>
  </datasetID>
  <name>Sample Part3 Dataset</name>
  <description>LandInfra dataset to test all possible content for Part3 Alignment
  </description>
  <dateTime>2016-12-02T10:00:00</dateTime>
  <datasetVersion>1.0</datasetVersion>
  <application>>manual</application>
  <author>Paul Scarponcini, Bentley Systems, Inc.</author>
  <infraVersion>1.0</infraVersion>
  <language>English</language>
  <defaultCRS xlink:href="crs1"/>
  <feature>
    <lia:Alignment gml:id="a1">
      <gml:description>sample alignment definition</gml:description>
      <gml:name>Road 1 centerline alignment</gml:name>
      <lia:alignmentID>
        <lia:ID>
          <identifier>Alignment1</identifier>
          <scope>OGC LandInfraSWG</scope>
        </lia:ID>
      </lia:alignmentID>
      <lia:purpose>roadway centerline</lia:purpose>
      <lia:designAlternative>DesignA</lia:designAlternative>
      <lia:linestring2DRepresentation>
```

```

<lia:Linestring2DRepresentation gml:id="l2D">
  <gml:pos>0 1000</gml:pos>
  <gml:pos>0 1100</gml:pos>
</lia:Linestring2DRepresentation>
</lia:linestring2DRepresentation>
<lia:linestring3DRepresentation>
  <lia:Linestring3DRepresentation gml:id="l3D">
    <gml:pos>0 1000 50</gml:pos>
    <gml:pos>0 1100 49</gml:pos>
  </lia:Linestring3DRepresentation>
</lia:linestring3DRepresentation>
<lia:geometry>
  <lia:AlignmentCurve gml:id="ac1">
    <lia:crs xlink:href="ecrs1"></lia:crs>
    <lia:horizontal>
      <lia:Alignment2DHorizontal gml:id="ah1">
        <lia:location>road centerline</lia:location>
        <lia:description>top finished surface</lia:description>
        <lia:state>proposed</lia:state>
        <lia:segment>
          <lia:Alignment2DHorSegment>
            <lia:tangentialContinuity>true</lia:tangentialContinuity>
            <lia:geometry>
              <lia:LineSegment>
                <gml:pos>0 1000</gml:pos>
                <gml:pos>0 1100</gml:pos>
              </lia:LineSegment>
            </lia:geometry>
          </lia:Alignment2DHorSegment>
        </lia:segment>
        <lia:segment>
          <lia:Alignment2DHorSegment>
            <lia:tangentialContinuity>true</lia:tangentialContinuity>
            <lia:geometry>
              <lia:ClothoidArcSegment>
                <gml:refLocation>
                  <gml:AffinePlacement>
                    <gml:location></gml:location>
                    <gml:refDirection></gml:refDirection>
                    <gml:inDimension>2</gml:inDimension>
                    <gml:outDimension>2</gml:outDimension>
                  </gml:AffinePlacement>
                </gml:refLocation>
                <gml:scaleFactor>1</gml:scaleFactor>
                <gml:startParameter>1</gml:startParameter>
                <gml:endParameter>1</gml:endParameter>
              </lia:ClothoidArcSegment>
            </lia:geometry>
          </lia:Alignment2DHorSegment>
        </lia:segment>
        <lia:segment>
          <lia:Alignment2DHorSegment>
            <lia:tangentialContinuity>true</lia:tangentialContinuity>
            <lia:geometry>
              <lia:CircularArcSegment>
                <lia:circularArcSegment>

```



```

        <lia:ArcByCenterPoint numArc="1">
          <gml:pos></gml:pos>
          <gml:radius uom="m">1</gml:radius>
          <gml:startAngle uom="d">1</gml:startAngle>
          <gml:endAngle uom="d">1</gml:endAngle>
        </lia:ArcByCenterPoint>
      </lia:circularArcSegment>
    </lia:ArcSegment>
  </lia:geometry>
</lia:Alignment2DHorSegment>
</lia:segment>
<lia:segment>
  <lia:Alignment2DHorSegment>
    <lia:tangentialContinuity>true</lia:tangentialContinuity>
    <lia:geometry>
      <lia:TransitionSegment gml:id="ts1">
        <lia:referenceLocation>
          <gml:AffinePlacement>
            <gml:location></gml:location>
            <gml:refDirection></gml:refDirection>
            <gml:inDimension>2</gml:inDimension>
            <gml:outDimension>2</gml:outDimension>
          </gml:AffinePlacement>
        </lia:referenceLocation>
        <lia:length uom="m">1</lia:length>
        <lia:startCurvature>1</lia:startCurvature>
        <lia:endCurvature>1</lia:endCurvature>
        <lia:transitionType>bloss</lia:transitionType>
      </lia:TransitionSegment>
    </lia:geometry>
  </lia:Alignment2DHorSegment>
</lia:segment>
</lia:Alignment2DHorizontal>
</lia:horizontal>
<lia:vertical>
  <lia:Alignment2DVertical gml:id="av1">
    <lia:location>top of finished pavement</lia:location>
    <lia:description>at road centerline</lia:description>
    <lia:state>proposed</lia:state>
    <lia:alignmentOffset uom="m">0</lia:alignmentOffset>
    <lia:segments>
      <lia:Alignment2DVertSegment>
        <lia:tangentialContinuity>true</lia:tangentialContinuity>
        <lia:startDistAlong>
          <gmlr:DistanceExpression gml:id="sda1">
            <gmlr:distanceAlong>1000</gmlr:distanceAlong>
          </gmlr:DistanceExpression>
        </lia:startDistAlong>
        <lia:startHeight uom="m">50</lia:startHeight>
        <lia:startGradient>-1.0</lia:startGradient>
        <lia:horizontalLength uom="m">100</lia:horizontalLength>
      </lia:Alignment2DVertSegment>
    </lia:segments>
  </lia:Alignment2DVertical>
</lia:vertical>

```

```

        <lia:startDistAlong>
          <gmlr:DistanceExpression gml:id="sda2">
            <gmlr:distanceAlong>1100</gmlr:distanceAlong>
          </gmlr:DistanceExpression>
        </lia:startDistAlong>
        <lia:startHeight uom="m">49</lia:startHeight>
        <lia:startGradient>-1.0</lia:startGradient>
        <lia:horizontalLength uom="m">100</lia:horizontalLength>
        <lia:isConvex>>false</lia:isConvex>
        <lia:constant>1</lia:constant>
        </lia:Alignment2DVertSegment>
      </lia:segments>
      <lia:measuredAlong xlink:href="ah1"></lia:measuredAlong>
    </lia:Alignment2DVertical>
  </lia:vertical>
</lia:AlignmentCurve>
</lia:geometry>
</lia:Alignment>
</feature>
<feature>
  <lia:Alignment gml:id="a2">
    <gml:description>another sample alignment definition</gml:description>
    <gml:name>Road 1 right EOP alignment</gml:name>
    <lia:alignmentID>
      <lia:ID>
        <identifier></identifier>
      </lia:ID>
    </lia:alignmentID>
    <lia:purpose>roadway right edge of pavement</lia:purpose>
    <lia:geometry>
      <lia:AlignmentCurve gml:id="ac2">
        <lia:horizontal xlink:href="ah1"/>
        <lia:vertical>
          <lia:Alignment2DVertical gml:id="av2">
            <lia:location>top of finished pavement</lia:location>
            <lia:description>at right edge of pavement</lia:description>
            <lia:state>proposed</lia:state>
            <lia:alignmentOffset uom="m">3.65</lia:alignmentOffset>
            <lia:segments>
              <lia:Alignment2DVertSegment>
                <lia:tangentialContinuity>>true</lia:tangentialContinuity>
                <lia:startDistAlong>
                  <gmlr:DistanceExpression gml:id="sda1-2">
                    <gmlr:distanceAlong>1000</gmlr:distanceAlong>
                  </gmlr:DistanceExpression>
                </lia:startDistAlong>
                <lia:startHeight uom="m">49.927</lia:startHeight>
                <lia:startGradient>-1.0</lia:startGradient>
                <lia:horizontalLength uom="m">100</lia:horizontalLength>
              </lia:Alignment2DVertSegment>
            </lia:segments>
            <lia:measuredAlong xlink:href="ah1"></lia:measuredAlong>
          </lia:Alignment2DVertical>
        </lia:vertical>
      </lia:AlignmentCurve>
    </lia:geometry>
  </lia:Alignment>
</feature>

```

```
</ia:Alignment>  
</feature>  
</LandInfraDataset>
```

## Annex C: Revision history

Date	Release	Author	Paragraph modified	Description
20170307	103r1	Scarponcini	Figure 2	Dependency changes in Part 6
20170410	103r2	Paul Scarponcini	Annex B	Fixed GML striping

## **Annex D: Bibliography**

- [1] OGC: OGC 08-131r3 The Specification Model — A Standard for Modular specifications, Open Geospatial Consortium, 2009