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/part0/1.0

Internal reference number of this OGC® document: 16-100r2

Version: 1.0

Category: OGC® Encoding Standard

Editor: Paul Scarponcini

OGC InfraGML 1.0: Part 0 – LandInfra Core - Encoding Standard

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

1

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

Figures

4

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 0 addresses the Core 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, survey, land parcel, land feature, terrain, road, rail, 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:

Name	Affiliation
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 0 of InfraGML addresses the following subject area(s): facilities, projects, alignment, road, railway, survey, land features, and land division. The InfraGML Core Requirements Class is included. This is the only mandatory InfraGML Requirements Class; every application claiming conformance to InfraGML must support it.

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[1].

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

---

1 www.opengeospatial.org/cite

7

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.

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:

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 or otherwise altered 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.

9

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/part0/1.0. 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 Land Division

Some InfraGML Parts depend upon other parts:

Copyright © 2017 Open Geospatial Consortium
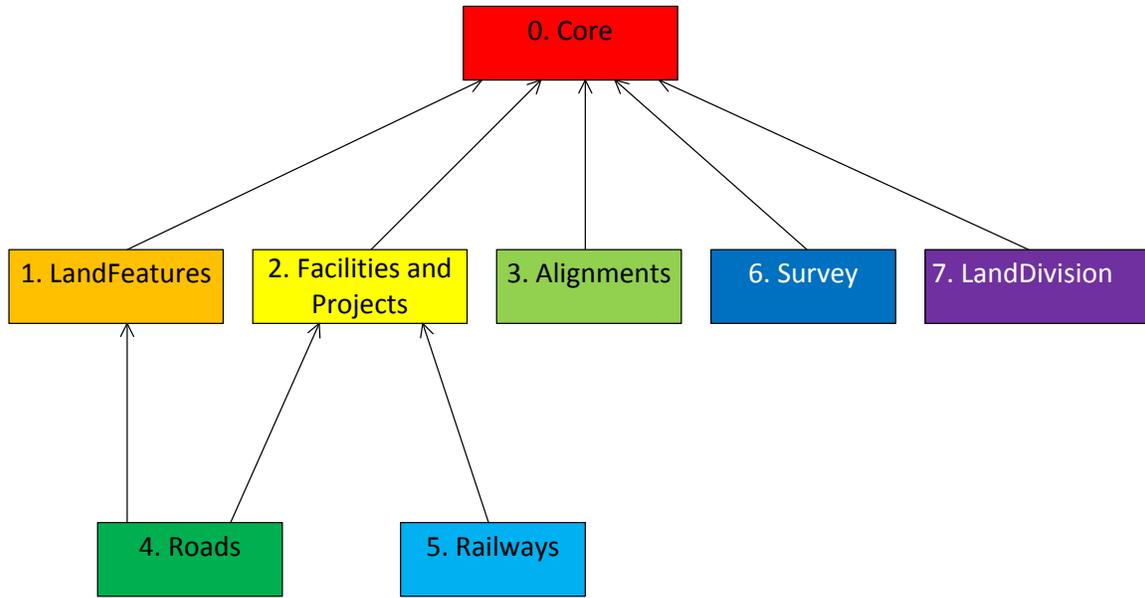
Figure 1. InfraGML Part Dependencies

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):

11
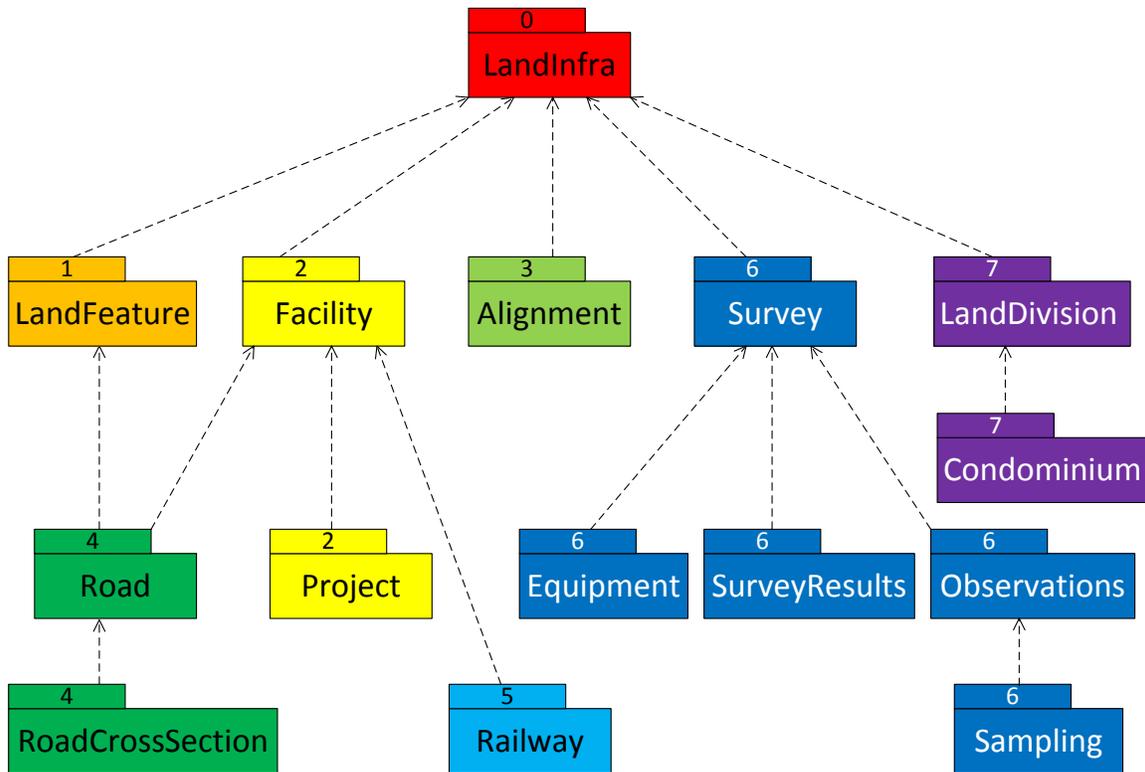
Figure 2. LandInfra Requirements Classes grouped into InfraGML Parts

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. 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 their Dependencies

7.1.1 Requirement Classes Defined in This Part

LandInfra Core

LandInfra Core 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 Core also contains the definition of types common across other Requirements Classes, such as the Status CodeList.
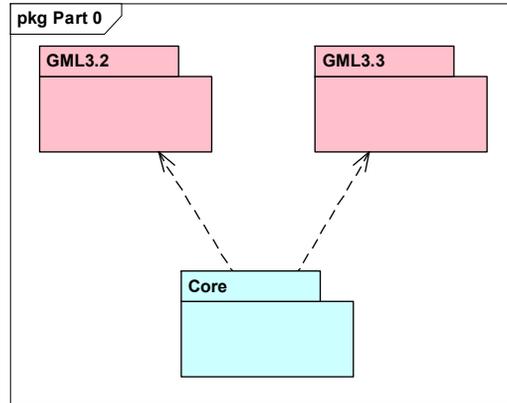
7.1.2 Other Standards upon which the Requirement Classes of this Part Depend

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: Core

Requirements Class
/req/core

13

Target type	Conforming application
Name	Core
Dependency	**http://www.w3.org/2001/XMLSchema**
Dependency	**http://www.opengis.net/gml/3.2**
Dependency	**http://www.opengis.net/gml/3.3/lr**
Dependency	**http://www.opengis.net/gml/3.3/lro**
Requirement	**/req/core/valid**
Requirement	**/req/core/header**
Requirement	**/req/core/gml3.2**
Requirement	**/req/core/gml3.3lr**
Requirement	**/req/core/gml3.3lro**
Requirement	**/req/core/elements**

7.2.1 Implementation decisions regarding OGC 15-111r1 UML

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

1. PolyfaceMesh has been changed to be a subtype of Geometry, instead of Brep Solid, as a PolyfaceMesh can be used to represent an (unclosed) surface as well as a (closed) 3D solid (See 7.2.3.1).

2. The type of the Document optional documentContent attribute has been changed from "FileName" to "anyURI".

3. In order for other InfraGML Parts to include their content under LandInfraDataset, an association between LandInfra::LandInfraDataset and LandInfra::Feature (0..*) is required. This works for all but Set, Survey::Survey and Facility::PositioningElement, as these are not Features. For these and other future non-Feature LandInfraDataset content, a new UML Class, AbstractData, is added. An association between LandInfra::LandInfraDataset and LandInfra::AbstractData (0..*) is also required. AbstractData can be extended by Property and PropertySet similar to Feature.

These changes are shown in the Logical Model in Figure 4.

4. Having featureID (identifier, optional scope) as an attribute of Feature leads to redundancies when Feature subTypes have their own ID (e.g., documentID, elementID). It has therefore been dropped as a Feature attribute. This may require subsequent changes in other Parts to Feature subtypes which do not already have their own ID.

5. As a result of the collaboration with BuildingSMART International, it was determined that ISO 19148 (OGC Abstract Specification Topic 19, Linear Referencing) did not completely address station equations. The agreed upon solution is to propose an additional Referent attribute called restartValue which behaves similar to the LinearElement.startValue and addresses the more general problem of restarting the measure value at the Referent location. This is achieved herein with the addition of a RestartReferent subtype of the original 19148 Referent, though an eventual change to

14

19148 itself (adding this optional attribute to Referent) may obviate the need for this extension, allowing the use of the revised gmllr:Referent instead. (see 7.2.3.2).

6. bSI introduced a way of specifying vector offset distance expressions without the need for the OGC Abstract Specification Topic 19 Vector Offset Requirements Class. An optional, longitudinal offset of type double (units defined by the LRM, positive direction in the direction of the linear element, bearing tangential to the linear element at the distance along point) has been added. Together with lateral and vertical offsets, this provides the same functionality as the vector offset without the added complexity. A new subtype of gmllro:LateralOffsetDistanceExpressionType called OffsetsDistanceExpression is added which extends it to include this optional longitudinal offset value.

7. OGC Abstract Specification Topic 19 (ISO 19148), Linear Referencing, requires that a linear element support the ILinearElement interface, instead of using subtyping. Because GML3.3 is a data standard, it added defaultLRM, measure, and startValue(s) properties along with a Feature, Curve, or Directed Edge to a LinearElement type. This is extended in this Part 0 Core to include optional Referents, which are owned by the linear element. See 7.2.3.3.

Since LinearElement is a subtype of gmllr:LinearElement, it cannot also be a Feature or an AbstractType and therefore must be added explicitly as a content type for LandInfraDataset. See Figure 4.

8. For all InfraGML Parts, types having a CodeList stereotype and containing a "codeList: URI" as the last value on the list are implemented in the XSD with type="gml:ReferenceType". In the XML, an xlink href specifies the source of the code list and a value from that code list optionally followed by an xlink title: xlink:href="http://example.com/professional#landSurveyor" xlink:title="Land Surveyor".

If no "codeList: URI" value is present, it is up to the Part to decide whether to implement the list as a ReferenceType or else as a union of an enumerated type plus a pattern type as specified in GML 3.2.1. However, if no "codeList: URI" value is present and xml document creators are not permitted to add to the list even though it has a stereotype of <<CodeList>> in the UML, the list of values should be encoded as if it were an enumeration.

9. For the Side enumeration, values of "neither" and "unspecified" have been added.

15

Figure 4. LandInfra Core Logical Model UML

7.2.2 General Requirements for all LandInfra Parts

Requirement	/req/core/valid
	Every XML instance document of any Part of InfraGML shall be valid according to the XML grammar specified in that Part for each Requirements Class contained in that Part and available at http://schemas.opengis.net/infragml/part\<n\>/1.0/\<schema name\>.xsd where \<n\> is the Part number and \<schema name\> is the name of the schema equal to the name of the respective Requirements Class.

The header for all specified schema provides namespaces and dependencies. The header for core processes below specifies the namespaces and dependent schema for GML 3.2 ("gml"), GML 3.3 Linear Referencing ("gmllr"), and GML 3.3 Linear Referencing Offsets ("gmlro") for Part 0 Core and GML 3.3 TIN ("gmltin") for Part 1 Land Features.

```xml
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:li="http://www.opengis.net/infragml/core/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:gmllr=" http://www.opengis.net/gml/3.3/lr"
xmlns:gmllro=" http://www.opengis.net/gml/3.3/lro"
xmlns:gmltin=" http://www.opengis.net/gml/3.3/tin"
targetNamespace="http://www.opengis.net/infragml/core/1.0"
elementFormDefault="qualified" version="1.0">
  <import namespace="http://www.opengis.net/gml/3.2"
    schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <import namespace="http://www.opengis.net/gml/3.3/lr"
    schemaLocation="http://schemas.opengis.net/gml/3.3/linearRef.xsd"/>
  <import namespace="http://www.opengis.net/gml/3.3/lro"
    schemaLocation="http://schemas.opengis.net/gml/3.3/linearRefOffset.xsd"/>
  <import namespace="http://www.opengis.net/gml/3.3/tin"
    schemaLocation="http://schemas.opengis.net/gml/3.3/tin.xsd"/>
```

Each InfraGML Requirements Class shall have its own XSD schema which shall contain a properly formed schema header appropriate to that Requirements Class and its dependencies. An InfraGML conforming application may modify the header so as to be able to include elements from multiple Requirements Classes (e.g., Road and Railway) in the same LandInfraDataset, so long as the schema header is still properly formed.

Requirement	/req/core/header
	An InfraGML conforming application shall use a schema definition file which shall contain a properly formed schema header appropriate to the InfraGML Requirements Classes it claims to support.

17

To be properly formed, the schema header for each InfraGML Requirements Class in every Part shall contain the following.

1. Namespace without prefix for xml of "http://www.w3.org/2001/XMLSchema"

2. Namespace for the Requirements Class of the form "http://www.opengis.net/infragml/<RC name>/1.0", for example "http://www.opengis.net/infragml/core/1.0" for the Core Requirements Class.

3. Namespace prefixes shall be defined for the targetNamespace and all dependent Requirement Class namespaces in accordance with Figure 5. For example, for the Core: xmlns:li="http://www.opengis.net/infragml/core/1.0".

4. All namespaces for InfraGML Requirements Classes upon which a dependency is declared shall have a schemaLocation of "http://schemas.opengis.net/infragml/part<n>/1.0/<RC name>.xsd" and be imported in the schema header. Even if namespace C is transitively dependent on namespace A via namespace B, then A, B, and C shall be included.

For inclusion of elements from additional InfraGML Requirements Classes supported by a conforming application, a properly formed schema header shall, in addition, include namespaces, namespace prefixes, and namespace inclusions for the additional InfraGML Requirements Classes and their dependencies. For example, an application supporting Road and Railway would use the Road XSD schema header augmented with the Railway namespace.

18

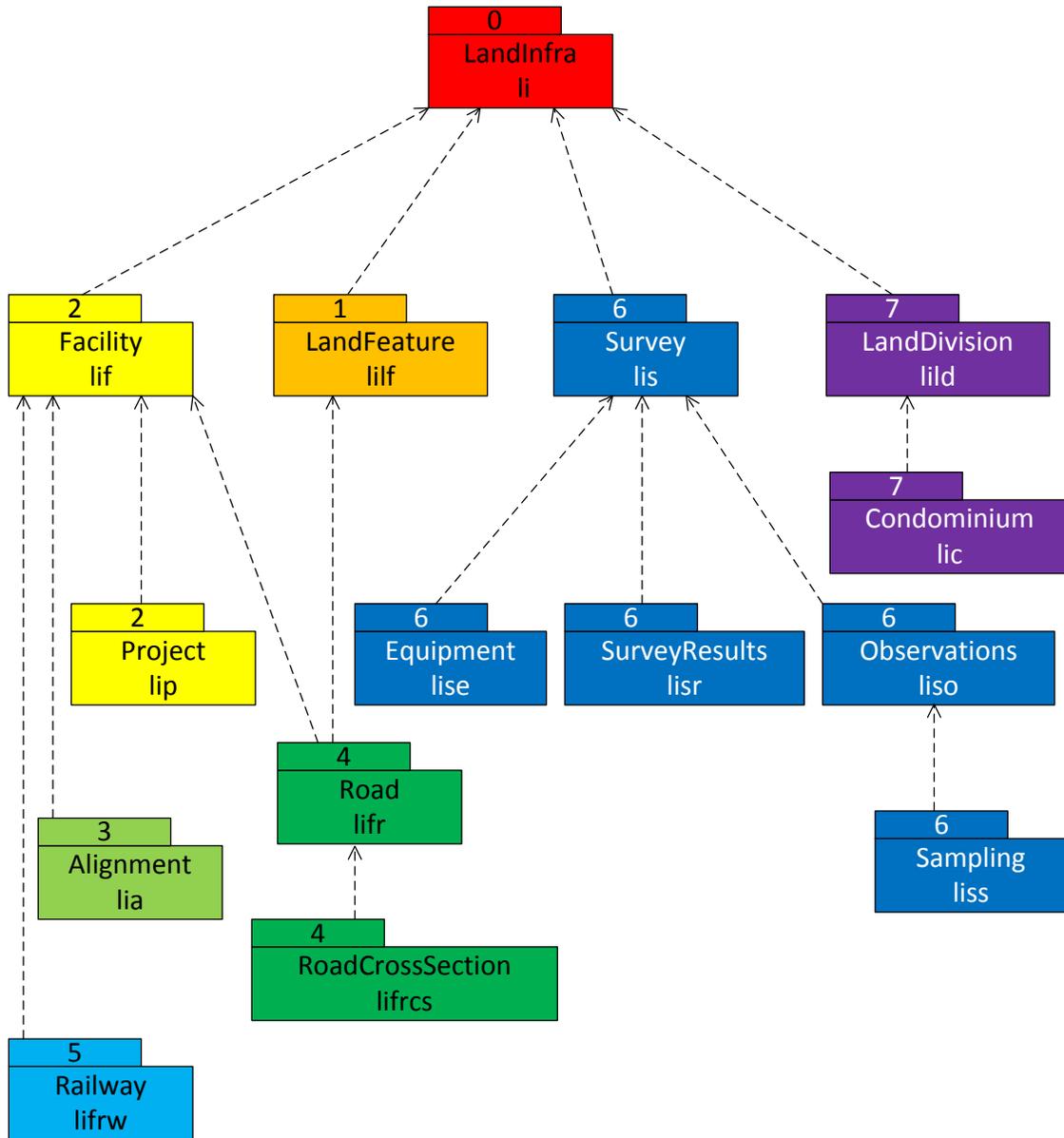*Figure 5. InfraGML Requirements Class namespace prefixes*

Requirement	/req/core/gml3.2
	This LandGML standard shall utilize and depend upon the OGC GML 3.2 standard schema within the http://www.opengis.net/gml/3.2 namespace identified by "gml".

19

Requirement	/req/core/gml3.3lr
	This LandGML standard shall utilize and depend upon the OGC GML 3.3 linearRef standard schema within the http://www.opengis.net/gml/3.3/lr namespace identified by "gmllr".

Requirement	/req/core/gml3.3lro
	This LandGML standard shall utilize and depend upon the OGC GML 3.3 linearRefOffset standard schema within the http://www.opengis.net/gml/3.3/lro namespace identified by "gmllro".

7.2.3 Specific Requirements for this Part

Requirement	/req/core/elements
	A conforming application shall support the Core XML elements listed in Table 1 in accordance with the GML XSD specified in http://schemas.opengis.net/infragml/part0/1.0/core.xsd.

An application conforming to this standard shall support the Core XML elements listed below in Table 1 in accordance with the GML XSD specified in http://schemas.opengis.net/infragml/part0/1.0/core.xsd. Core XML element names are shown with a XML namespace prefix of "li". Corresponding LandInfra UML classes are shown with their LandInfra Requirements Class prefix of "LandInfra".

InfraGML Core XML element	LandInfra UML Class
li:AbstractData	InfraGML Core::AbstractData
li:Document	LandInfra::Document
li:Feature	LandInfra::Feature
li:FeatureAssociation	LandInfra::FeatureAssociation
li:IndexedPoint	InfraGML Core::IndexedPoint
li:IndexedPointList	
li:LandInfraDataset	LandInfra::LandInfraDataset
li:LinearElement	
li:OffsetsDistanceExpression	InfraGML Core::OffsetsDistanceExpression

20

li:Percentage	LandInfra::Percentage
li:PolyfaceMesh	InfraGML Core::PolyfaceMesh
li:Professional	LandInfra::Professional
li:Property	LandInfra::Property
li:PropertySet	LandInfra::PropertySet
li:RestartReferent	LandInfra Core::RestartReferent
li:Set	LandInfra::Set
li:SideType	LandInfra::Side
li:SimpleIndexedPolygon	InfraGML Core::SimpleIndexedPolygon
li:SimpleIndexedPolygonList	
li:StateType	LandInfra::StateType
li:SurveyMark	LandInfra::SurveyMark

Table 1. InfraGML Core XML elements with corresponding LandInfra UML classes

7.2.3.1 Polyface Mesh

A PolyfaceMesh is a geometry type often used to define the Spatial Representation of Physical Elements such as Road and Railway Elements. A PolyfaceMesh consists of one or more SimpleIndexedPolygons.   A SimpleIndexedPolygon has at least 3 sides. These SimpleIndexedPolygons are defined by a list of at least three Index values. Each Index value represents an IndexedPoint which contains the index and a list of coordinates. The list of eligible IndexedPoints used by a PolyfaceMesh is contained in the PolyfaceMesh definition.   PolyfaceMesh is not defined in the OGC Abstract Specification Topic 1, Geometry, and therefore its UML Conceptual Model is shown below as Figure 6.

Note that, unlike in OGC 15-111r1, PolyfaceMesh is a subtype of Geometry, not Brep Solid, as a PolyfaceMesh can be used to represent an (unclosed) surface as well as a (closed) 3D solid. For a closed solid, the point indices within each polygon should be ordered so that they proceed in a counterclockwise direction when viewed from the outside of the closed volume.

For an unclosed mesh, one "side" of the overall surface should be considered the positive side, and all point indices should be ordered so they proceed counterclockwise around the face when viewed from the positive side.

Note that with this ordering, when an edge is shared by two facets its two vertex indices will appear in reversed order within the lists of the two facets.   In the example below in Figure 7, look at the edge with vertices 6 and 9. The (5 6 9) triangle "uses" this edge with the order (6 9), and the (6 7 9) triangle "uses" this edge with the order (9 6).
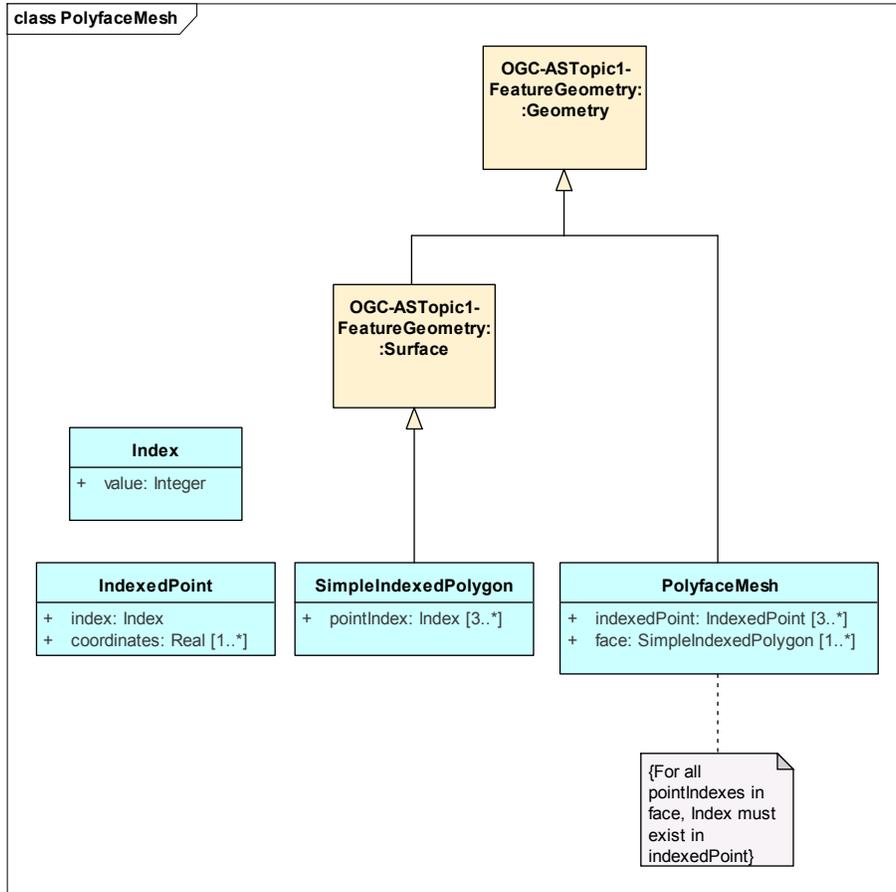
Figure 6. PolyfaceMesh Conceptual Model

Because Polyface Mesh is not included in OGC Topic 1, it is also not GML encoded in GML3.2. The GML encoding is therefore defined in this InfraGML Part 0 and (informatively) shown here:

```xml
<!--                                                                          -->
<!--     Indexed Point                                                        -->
<!--                                                                          -->
<element name="IndexedPoint" type="li:IndexedPointType" >
  <annotation>
    <documentation>Point having an Integer index reference number</documentation>
  </annotation>
</element>
<complexType name="IndexedPointType">
  <sequence>
    <element name="index" type="integer"/>
    <element name="coordinates" type="gml:doubleList"/>
  </sequence>
</complexType>
<!--                                                                          -->
<!--     Indexed Point List                                                   -->
<!--                                                                          -->
<element name="IndexedPointList" type="li:IndexedPointListType" />
<complexType name="IndexedPointListType">
```

22

```xml
    <sequence>
      <element ref="li:IndexedPoint" minOccurs="3" maxOccurs="unbounded">
      </element>
    </sequence>
  </complexType>
  <!--                                                                          -->
  <!--     Polyface Mesh                                                        -->
  <!--                                                                          -->
  <element name="PolyfaceMesh" type="li:PolyfaceMeshType"
substitutionGroup="gml:AbstractGeometry">
    <annotation>
      <documentation>A PolyfaceMesh is a Solid bounded by simple planar faces such as
simple triangles or quadrilaterals.  For compactness the faces are defined by a list of integers
representing Point indexes where the IndexedPoints have also been  defined with their index and
coordinate values.</documentation>
    </annotation>
  </element>
  <complexType name="PolyfaceMeshType">
    <complexContent>
      <extension base="gml:AbstractGeometryType">
        <sequence>
          <element ref="li:IndexedPointList"/>
          <element ref="li:SimpleIndexedPolygonList"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!--                                                                          -->
  <!--     Simple Indexed Polygon                                              -->
  <!--                                                                          -->
  <element name="SimpleIndexedPolygon" type="li:SimpleIndexedPolygonType"
substitutionGroup="gml:AbstractSurface">
    <annotation>
      <documentation>Simple Polygon Surface defined by IndexedPoint
indices</documentation>
    </annotation>
  </element>
  <complexType name="SimpleIndexedPolygonType">
    <complexContent>
      <extension base="gml:AbstractSurfaceType">
        <sequence>
          <element name="pointIndex" type="gml:integerList"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!--                                                                          -->
  <!--     Simple Indexed Polygon List                                         -->
  <!--                                                                          -->
  <element name="SimpleIndexedPolygonList" type="li:SimpleIndexedPolygonListType" />
  <complexType name="SimpleIndexedPolygonListType">
    <sequence>
      <element ref="li:SimpleIndexedPolygon" minOccurs="1" maxOccurs="unbounded">
      </element>
    </sequence>
```

23

```
  </complexType>
</schema>
```

Consider the following (informative) example:

Figure 7. PolyfaceMesh example

The resultant GML defining a PolyfaceMesh for just the pyramidal top would be:

```
<PolyfaceMesh gml:id="M1">
  <IndexedPointList>
    <IndexedPoint>
      <index>5</index>
      <coordinates>0 0 2</coordinates>
    </IndexedPoint>
    <IndexedPoint>
      <index>6</index>
      <coordinates>2 0 2</coordinates>
    </IndexedPoint>
    <IndexedPoint>
      <index>7</index>
      <coordinates>2 2 2</coordinates>
    </IndexedPoint>
    <IndexedPoint>
      <index>8</index>
      <coordinates>0 2 2</coordinates>
    </IndexedPoint>
    <IndexedPoint>
      <index>9</index>
      <coordinates>1 1 3</coordinates>
    </IndexedPoint>
  </IndexedPointList>
  <SimpleIndexedPolygonList>
    <SimpleIndexedPolygon gml:id="f1">
      <pointIndex>5 8 7 6</pointIndex>
    </SimpleIndexedPolygon>
    <SimpleIndexedPolygon gml:id="f2">
      <pointIndex>5 6 9</pointIndex>
    </SimpleIndexedPolygon>
    <SimpleIndexedPolygon gml:id="f3">
      <pointIndex>6 7 9</pointIndex>
```
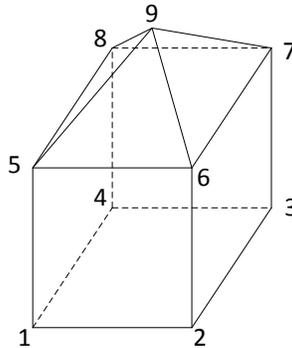
```
      </SimpleIndexedPolygon>
      <SimpleIndexedPolygon gml:id="f4">
         <pointIndex>7 8 9</pointIndex>
      </SimpleIndexedPolygon>
      <SimpleIndexedPolygon gml:id="f5">
         <pointIndex>8 5 9</pointIndex>
      </SimpleIndexedPolygon>
   </SimpleIndexedPolygonList>
</PolyfaceMesh>
```

Note that if the rectangular solid base was to be included in the same PolyfaceMesh, IndexedPoints 1, 2, 3, and 4 would also have to be included in the mesh definition. If the rectangular solid base was to be defined as a separate geometry, a separate PolyfaceMesh would have to be defined, including IndexedPoints 1, 2, 3, 4, 5, 6, 7, and 8.

7.2.3.2 Restart Referent

An extension to OGC Abstract Specification Topic 19, Linear Referencing (ISO 19148), this subtype of the Topic 19 Referent type allows for the specification of a restart measure value at the Referent location for use by all linearly referenced locations which include this Referent and use the specified LRM, similar to the startValue of a LinearElement. Together with the Referent Location, this optional restart value (of type StartValue from Topic 19) can be used to specify a station equation, though the approach taken solves the more general case where the measuring needs to be restarted with a value other than zero at the Referent location.
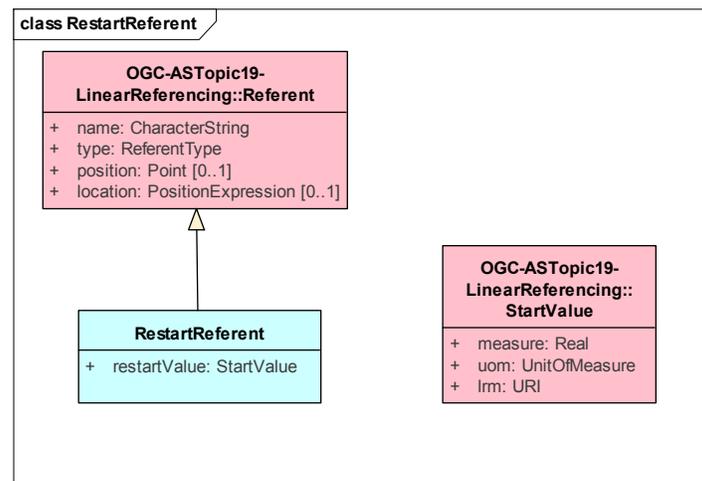

Figure 8. RestartReferent UML

The (informative) example in Figure 9 shows a station equation along Alignment AL1 to be encoded as RestartReferent RefA. The station equation is 3+50 Back = 4+00 Ahead, marking a discontinuity in the stationing along AL1. Station values prior to the station

25

equation (Back) are measured from the start station of 2+50. Station values after to the station equation (Ahead) are measured from the restart station of 4+00.
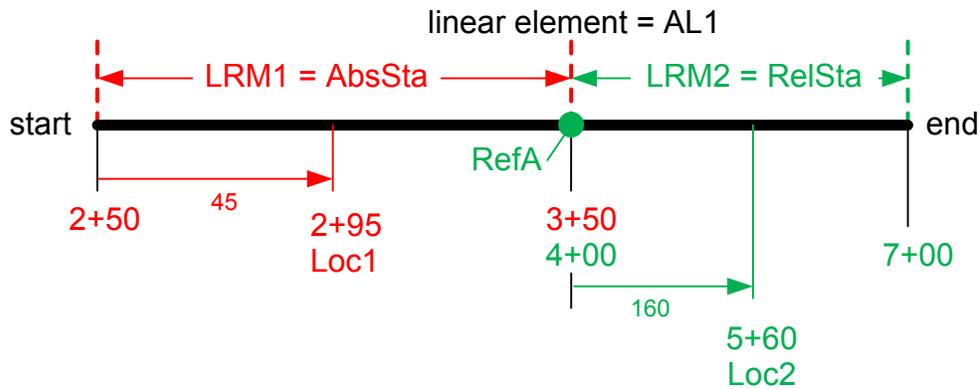
Figure 9. Station Equation example

Linear element AL1 has a default LRM (LRM1) of absolute stationing, a total length (measure) of 400 feet, and a start value of 2+50 when using LRM1:

```
<LinearElement gml:id="AL1">
  <feature/>
  <defaultLRM>
    <LinearReferencingMethod gml:id="LRM1">
      <name>AbsoluteStationing</name>
      <type>absolute</type>
      <units>feet</units>
    </LinearReferencingMethod>
  </defaultLRM>
  <measure uom="feet">400</measure>
  <startValue lrm="LRM1" uom="feet">250</startValue>
</LinearElement>
```

The first location, Loc1, has a linearly referenced location along AL1 of 2+95, or 45 feet measured from the start of AL1, using AL1's default LRM of LRM1:

```
<PositionExpression gml:id="peLoc1">
  <linearElement>
    <LinearElement gml:id="AL1">
    </LinearElement>
  </linearElement>
  <lrm/>
  <distanceExpression>
    <DistanceExpression gml:id="deLoc1">
      <distanceAlong>295</distanceAlong>
    </DistanceExpression>
  </distanceExpression>
</PositionExpression>
```

26

The station equation is encoded as a RestartReferent called RefA. The referentName is "station equation". RefA is located using LRM1, that is using absolute stationing along AL1 from its start, adjusted for the AL1/LRM1 start value of 2+50. Being 100 feet from the start of AL1 translates into an absolute station of 3+50. For subsequent station measures along AL1 beyond RefA, the measures are adjusted by the restart value of 4+00 (the Ahead station value).

```xml
<RestartReferent gml:id="RefA">
  <gmllr:name>station equation</gmllr:name>
  <gmllr:type>boundary</gmllr:type>
  <gmllr:location>
    <gmllr:PositionExpression gml:id="peRefA">
      <gmllr:linearElement xlink:href="AL1"></gmllr:linearElement>
      <gmllr:lrm xlink:href="LRM1"></gmllr:lrm>
      <gmllr:distanceExpression>
        <gmllr:DistanceExpression gml:id="deRefA">
          <gmllr:distanceAlong>350</gmllr:distanceAlong>
        </gmllr:DistanceExpression>
      </gmllr:distanceExpression>
    </gmllr:PositionExpression>
  </gmllr:location>
  <gmllr:ownedBy xlink:href="AL1"></gmllr:ownedBy>
  <restartValue lrm="LRM2">400</restartValue>
</RestartReferent>
```

The second location, Loc2, has a linearly referenced location along AL1 of 5+60, based on the Ahead stationing. That would make it 160 feet from the station equation at RefA and 260 feet from the start of AL1.

```xml
<PositionExpression gml:id="peLoc2">
  <linearElement>
    <LinearElement gml:id="AL1">
    </LinearElement>
  </linearElement>
  <lrm>
    <LinearReferencingMethod gml:id="LRM2">
      <name>RelativeStationing</name>
      <type>relative</type>
      <units>feet</units>
    </LinearReferencingMethod>
  </lrm>
  <distanceExpression>
    <DistanceExpression gml:id="deLoc2">
      <distanceAlong>560</distanceAlong>
      <referent>
        <AlongReferent gml:id="ar1">
          <fromReferent>
            <Referent gml:id="RefA"></Referent>
          </fromReferent>
        </AlongReferent>
      </referent>
    </DistanceExpression>
  </distanceExpression>
```

```
</PositionExpression>
```

7.2.3.3 LinearElement

OGC Abstract Specification Topic 19 (ISO 19148), Linear Referencing, allows any Feature, Curve, or DirectedEdge that supports the ILinearElement interface to act as a linear element along which linearly referenced locations can be defined. For a data standard such as GML3.3, this was achieved by adding defaultLRM, measure, and startValue(s) properties along with the Feature, Curve, or Directed Edge to a LinearElement type. This is extended in this Part 0 Core to include optional Referents, which are owned by the linear element, giving LinearElement the following attributes:

> choice of:

>> **feature:** the Feature behaving as a linear element;

>> **curve:** the Curve behaving as a linear element; or

>> **edge:** the DirectedEdge behaving as a linear element;

> **defaultLRM**: the Linear Referencing Method to be used for measurements along this linear element unless specified otherwise;

> **measure**: the overall length (or weight, if a DirectedEdge) of the linear element;

> **startValue**: any number of startValue measurements; and

> **referent**: any number of Referents owned by the linear element.

If an LRM supported by the LinearElement is of type "absolute", then the measured value at the start of the LinearElement, if other than 0, is specified by the startValue. A startValue is specified as a decimal number attributed by an lrm and units values. See OGC Abstract Specification Topic 19 (ISO 19148), Linear Referencing, for greater explanation of any of these terms.

LandInfra PositioningElement is not implemented in InfraGML. Alignment is changed to being a subtype of Feature in Part 3 Alignment. This allows an Alignment to behave as a linear element.

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: Core

/conf/core		
Requirements	/req/core	
Dependency		
Test	/conf/core/valid	
	Requirement	/req/core/valid
	Test purpose	Verify that XML instance documents of any Part of InfraGML is valid according to the XML grammar specified in that Part for each Requirements Class contained in that Part and available at http://schemas.opengis.net/infragml/part\<n\>/1.0/\<schema name\>.xsd where \<n\> is the Part number and \<schema name\> is the name of the schema equal to the name of the respective Requirements Class.
	Test method	Validate the XML document using the XML schema document(s) for the applicable Part(s). Pass if no errors reported. Fail otherwise.
	Test type	Basic
Test	/conf/core/header	
	Requirement	/req/core/header
	Test purpose	Verify that the conforming application uses a schema definition file which contains a properly formed schema header appropriate to the InfraGML Requirements Classes it claims to support.
	Test method	Inspect the schema definition file used by the application.
	Test type	Basic
Test	/conf/core/gml3.2	
	Requirement	/req/core/gml3.2
	Test purpose	Verify that the conforming application utilizes and depends upon the OGC GML 3.2 standard schema within the http://www.opengis.net/gml/3.2 namespace identified by "gml".
	Test method	Inspect the GML output to verify that all appropriate elements have a "gml:" prefix.
	Test type	Capability
Test	/conf/core/gml3.3lr	
	Requirement	/req/core/gml3.3lr
	Test purpose	Verify that the conforming application utilizes and depends upon the OGC GML 3.3 linearRef standard schema within the http://www.opengis.net/gml/3.3/lr namespace identified by "gmllr".

29

	Test method	Inspect the GML output to verify that all appropriate elements have a "gmllr:" prefix.
	Test type	Capability
Test	**/conf/core/gml3.3lro**	
	Requirement	**/req/core/gml3.3lro**
	Test purpose	Verify that the conforming application utilizes and depends upon the OGC GML 3.3 linearRefOffset standard schema within the http://www.opengis.net/gml/3.3/lro namespace identified by "gmllro".
	Test method	Inspect the GML output to verify that all appropriate elements have a "gmllro:" prefix.
	Test type	Capability
Test	**/conf/core/elements**	
	Requirement	**/req/core/elements**
	Test purpose	Verify that the conforming application supports the Core XML elements listed in Table 1 in accordance with the GML XSD specified in http://schemas.opengis.net/infragml/part0/1.0/core.xsd.
	Test method	Inspect the GML output to verify the above requirement.
	Test type	Capability

30

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 Core XML

Example from Part0Core0410.xsd

```xml
<?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="ds0"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gmllr="http://www.opengis.net/gml/3.3/lr"
xmlns:gmllro="http://www.opengis.net/gml/3.3/lro"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/infragml/core/1.0 Part0Core0410.xsd">
  <datasetID>
    <ID>
      <identifier>DS0</identifier>
      <scope>OGC LandInfraSWG</scope>
    </ID>
  </datasetID>
  <name>Sample Part0 Dataset</name>
  <description>LandInfra dataset to test all possible content for Part0 Core</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>
    <Feature gml:id="f1">
      <spatialRepresentation>
        <SpatialRepresentation>
          <geometry>
            <gml:Point gml:id="p2">
              <gml:pos>0 0</gml:pos>
            </gml:Point>
          </geometry>
        </SpatialRepresentation>
      </spatialRepresentation>
      <linearlyReferencedLocation>
        <LinearlyReferencedLocation>
          <linearAtLocation>
            <LinearAtLocation>
              <atPosition>
                <gmllr:PositionExpression gml:id="pe1">
                  <gmllr:linearElement>
                    <LinearElement gml:id="AL11">
```

31

```xml
<gmllr:feature xlink:href="f2"></gmllr:feature>
<gmllr:defaultLRM>
  <gmllr:LinearReferencingMethod gml:id="lrm0">
    <gmllr:name>chainage</gmllr:name>
    <gmllr:type>absolute</gmllr:type>
    <gmllr:units>m</gmllr:units>
  </gmllr:LinearReferencingMethod>
</gmllr:defaultLRM>
<gmllr:measure uom="m">10</gmllr:measure>
<gmllr:startValue lrm="lrm1" uom="m">100</gmllr:startValue>
<referent>
  <RestartReferent gml:id="RefA">
    <gmllr:name>station equation</gmllr:name>
    <gmllr:type>boundary</gmllr:type>
    <gmllr:location>
      <gmllr:PositionExpression gml:id="peRefA">
        <gmllr:linearElement xlink:href="AL1" xlink:title="LinearElement">
        </gmllr:linearElement>
        <gmllr:lrm xlink:href="LRM1"></gmllr:lrm>
        <gmllr:distanceExpression>
          <gmllr:DistanceExpression gml:id="deRefA">
            <gmllr:distanceAlong>350</gmllr:distanceAlong>
          </gmllr:DistanceExpression>
        </gmllr:distanceExpression>
      </gmllr:PositionExpression>
    </gmllr:location>
    <gmllr:ownedBy xlink:href="AL1"></gmllr:ownedBy>
    <restartValue lrm="LRM2">400</restartValue>
  </RestartReferent>
</referent>
            </LinearElement>
          </gmllr:linearElement>
          <gmllr:lrm xlink:href="lrm1"></gmllr:lrm>
          <gmllr:distanceExpression>
            <gmllr:DistanceExpression gml:id="de1">
              <gmllr:distanceAlong>3</gmllr:distanceAlong>
            </gmllr:DistanceExpression>
          </gmllr:distanceExpression>
        </gmllr:PositionExpression>
      </atPosition>
    </LinearAtLocation>
  </linearAtLocation>
</LinearlyReferencedLocation>
</linearlyReferencedLocation>
</Feature>
</feature>
<feature>
  <Feature gml:id="f2">
    <spatialRepresentation>
      <SpatialRepresentation>
        <geometry xlink:href="p1">
        </geometry>
      </SpatialRepresentation>
    </spatialRepresentation>
    <linearlyReferencedLocation>
      <LinearlyReferencedLocation>
```

```xml
              <linearAtLocation>
                <LinearAtLocation>
                  <atPosition xlink:href="pe1">
                  </atPosition>
                </LinearAtLocation>
              </linearAtLocation>
            </LinearlyReferencedLocation>
          </linearlyReferencedLocation>
        </Feature>
    </feature>
    <feature>
        <Feature gml:id="f3">
          <gml:name>ELR</gml:name>
          <propertySet>
            <PropertySet gml:id="ps1">
              <gml:description>properties from 19148 ILinearElemnt</gml:description>
              <gml:name>LinearElement</gml:name>
              <authority>OGC/ISO</authority>
              <propertyMember>
                <Property gml:id="prop1">
                  <gml:description>LRM used for a linear element when none is
                    specified</gml:description>
                  <gml:name>defaultLRM</gml:name>
                  <valueType>string</valueType>
                  <value>milepost</value>
                  <units>miles</units>
                  <cardinality>1</cardinality>
                </Property>
              </propertyMember>
            </PropertySet>
          </propertySet>
        </Feature>
    </feature>
    <feature>
        <Feature gml:id="f4">
          <spatialRepresentation>
            <SpatialRepresentation>
              <geometry>
                <PolyfaceMesh gml:id="pm1">
                  <indexedPointList>
                    <IndexedPoint>
                      <index>1</index>
                      <coordinates>0 0 0</coordinates>
                    </IndexedPoint>
                    <IndexedPoint>
                      <index>2</index>
                      <coordinates>1 0 0</coordinates>
                    </IndexedPoint>
                    <IndexedPoint>
                      <index>3</index>
                      <coordinates>1 1 0</coordinates>
                    </IndexedPoint>
                  </indexedPointList>
                  <simpleIndexedPolygonList>
                    <SimpleIndexedPolygon gml:id="spl1">
                      <pointIndex>1 2 3</pointIndex>
```

33

```xml
                    </SimpleIndexedPolygon>
                  </simpleIndexedPolygonList>
                </PolyfaceMesh>
              </geometry>
            </SpatialRepresentation>
          </spatialRepresentation>
        </Feature>
      </feature>
      <linearElement>
        <LinearElement gml:id="le1">
          <gmllr:feature xlink:href="f3"></gmllr:feature>
          <gmllr:defaultLRM xlink:href="lrm1">
          </gmllr:defaultLRM>
          <gmllr:measure uom="m">100</gmllr:measure>
          <gmllr:startValue lrm="lrm1">0</gmllr:startValue>
          <referent>
            <gmllr:Referent gml:id="r1">
              <gmllr:name>Dox County boundary</gmllr:name>
              <gmllr:type>boundary</gmllr:type>
              <gmllr:ownedBy xlink:href="le1">
              </gmllr:ownedBy>
            </gmllr:Referent>
          </referent>
        </LinearElement>
      </linearElement>
      <feature>
        <SurveyMark gml:id="sm1">
          <spatialRepresentation>
            <SpatialRepresentation>
              <geometry>
                <gml:Point gml:id="p1">
                  <gml:pos>105 230</gml:pos>
                </gml:Point>
              </geometry>
            </SpatialRepresentation>
          </spatialRepresentation>
          <identification>1001</identification>
        </SurveyMark>
      </feature>
      <featureAssociation>
        <FeatureAssociation gml:id="fa1">
          <gml:description>associates document 20160004 with Survey Mark
            1001</gml:description>
          <gml:name>survey marking</gml:name>
          <fromFeature xlink:href="d1"/>
          <fromRole>documentation for</fromRole>
          <toFeature xlink:href="sm1"></toFeature>
          <toRole>documented survey mark</toRole>
        </FeatureAssociation>
      </featureAssociation>
      <feature>
        <Document gml:id="d1">
          <documentID>
            <ID>
              <identifier>20160004</identifier>
            </ID>
```

```
            </documentID>
            <documentType>Survey Marking</documentType>
            <documentContent>http://city.net/survey/documents.pdf"</documentContent>
        </Document>
    </feature>
</LandInfraDataset>
```

Annex C: Revision history

Date	Release	Author	Paragraph modified	Description
20170321	100r1	Paul Scarponcini	Figures 2 and 5	RC dependency changes in Part 6
20170321	100r1	Paul Scarponcini	7.2.1 - 8	Code List encoding rules modified
20170321	100r1	Paul Scarponcini	Figure 4	Removed redundant LandInfraDataset-Set association
20170410	100r2	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