

# Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: 2022-04-11

External identifier of this OGC® document: <http://www.opengis.net/doc/dp/DynamicFeatures/{m.n}>

Internal reference number of this OGC® document: 22-005

Category: OGC® Discussion Paper

Editor: Charles Heazel

## OGC Dynamic Features Discussion Paper

### Copyright notice

Copyright © 2022 Open Geospatial Consortium

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

### Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Document type: OGC® Discussion Paper

Document subtype:

Document stage: Draft

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.

# Table of Contents

1. Scope .....	6
2. References .....	7
3. Terms and Definitions .....	8
3.1. <b>base representation</b> .....	8
3.2. <b>design coordinate reference system</b> .....	8
3.3. <b>direct position</b> .....	8
3.4. <b>external coordinate reference system</b> .....	8
3.5. <b>feature</b> .....	8
3.6. <b>foliation</b> .....	8
3.7. <b>geometric object</b> .....	9
3.8. <b>geometric primitive</b> .....	9
3.9. <b>global coordinate system</b> .....	9
3.10. <b>GM_Point</b> .....	9
3.11. <b>instant</b> .....	9
3.12. <b>leaf</b> .....	9
3.13. <b>one parameter set of geometries</b> .....	9
3.14. <b>prism</b> .....	9
3.15. <b>temporal coordinate system</b> .....	10
3.16. <b>temporal position</b> .....	10
3.17. <b>temporal reference system</b> .....	10
3.18. <b>trajectory</b> .....	10
3.19. <b>vector</b> .....	10
4. Conventions .....	11
4.1. <b>Identifiers</b> .....	11
5. Dynamic Features .....	12
5.1. <b>Geometry in 3 Dimensions</b> .....	12
5.1.1. <b>Feature Model</b> .....	12
5.1.2. <b>Feature Geometry</b> .....	13
5.1.3. <b>3D Geometry</b> .....	16
5.2. <b>Moving Features</b> .....	21
5.2.1. <b>Coordinate systems</b> .....	22
5.2.2. <b>Coordinate Representation</b> .....	24
5.2.3. <b>Geometry</b> .....	26
5.2.4. <b>Location</b> .....	27
5.2.5. <b>Orientation</b> .....	29
5.2.6. <b>Non-rigid Bodies</b> .....	32
5.3. <b>Articulated Geometries</b> .....	32
5.3.1. <b>GeoPOSE</b> .....	32

5.3.2. MISB Staging System .....	33
5.3.3. Discussions .....	35
5.4. Mass Properties .....	35
5.5. Temporal Reference Systems .....	35
5.5.1. MISB Timers .....	35
5.5.2. Temporal Schema .....	37
5.6. Dynamic Features at Relativistic Velocities .....	37
5.6.1. Lorentz Transformation .....	38
5.6.2. Minkowski Space Time .....	38
Annex A: Bibliography .....	40
Annex B: Revision History .....	41

## i. Abstract

Moving Feature standards and technologies have made considerable progress. It is time to evaluate the foundations of OGC standards to determine if they are sufficient to support current and future Moving Feature standards. This discussion paper examines ISO and OGC standards against current and anticipated future needs. It also examines other standards and conventions which may be useful for enhancing the OGC/ISO foundation.>

## ii. Keywords

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

ogcdoc, OGC document, <tags separated by commas>

## iii. Preface

### NOTE

Insert Preface Text here. Give OGC specific commentary: describe the technical content, reason for document, history of the document and precursors, and plans for future work. > 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):

Organization name(s)

## v. Submitters

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

Name Affiliation

# Chapter 1. Scope

Moving Feature standards and technologies have made considerable progress. It is time to evaluate the foundations of OGC standards to determine if they are sufficient to support current and future Moving Feature standards. This discussion paper examines ISO and OGC standards against current and anticipated future needs. It also examines other standards and conventions which may be useful for enhancing the OGC/ISO foundation.

# Chapter 2. References

As a Discussion Paper, this document contains no normative references. Citations for resources referenced in this document are listed in the [Bibliography](#) (Annex B).

# Chapter 3. 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 Best Practice.

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

## 3.1. base representation

<moving features>representation, using a local origin and local ordinate vectors, of a geometric object at a given reference time [ISO 19141:2008, definition 4.1.1]

NOTE 1 A rigid geometric object may undergo translation or rotation, but remains congruent with its base representation.

NOTE 2 The local origin and ordinate vectors establish an engineering coordinate reference system (ISO 19111), also called a local frame or a local Euclidean coordinate system.

## 3.2. design coordinate reference system

engineering coordinate reference system in which the base representation of a moving object is specified. [ISO 19141:2008, definition 4.1.3]

## 3.3. direct position

DirectPosition object data types hold the coordinates for a position within some coordinate reference system.

## 3.4. external coordinate reference system

the coordinate reference system in which the geometry of the curve for a trajectory is defined. Usually an earth-fixed geographic CRS. [ISO 19141:2008, derived]

## 3.5. feature

abstraction of real world phenomena [ISO 19101:2002, definition 4.11]

NOTE A feature may occur as a type or an instance. Feature type or feature instance shall be used when only one is meant.

## 3.6. foliation

one parameter set of geometries such that each point in the prism of the set is in one and only one trajectory and in one and only one leaf [ISO 19141:2008, definition 4.1.8]



## 3.7. geometric object

spatial object representing a geometric set [ISO 19107:2003, definition 4.47]

## 3.8. geometric primitive

geometric object representing a single, connected, homogeneous element of space [ISO 19107:2003, definition 4.48]

NOTE Geometric primitives are non-decomposed objects that present information about geometric configuration. They include points, curves, surfaces, and solids.

## 3.9. global coordinate system

the CRS of the trajectory of the reference point.

NOTE A global coordinate system is usually in terms of the moving frame of the curve (oriented on the local tangent) or in terms of the external CRS in which the geometry of the curve is defined. [ISO 19141:2008, derived]

## 3.10. GM\_Point

GM\_Point is the basic data type for a geometric object consisting of one and only one point. That point is represented as a DirectPosition.

## 3.11. instant

0-dimensional geometric primitive representing position in time [ISO 19108:2002, definition 4.1.17]

## 3.12. leaf

<one parameter set of geometries> geometry at a particular value of the parameter [ISO 19141:2008, definition 4.1.12]

## 3.13. one parameter set of geometries

function  $f$  from an interval  $t \in [a, b]$  such that  $f(t)$  is a geometry and for each point  $P \in f(a)$  there is a one parameter set of points (called the trajectory of  $P$ )  $P(t) : [a, b] \rightarrow P(t)$  such that  $P(t) \in f(t)$  [ISO 19141:2008, definition 4.1.15]

EXAMPLE A curve  $C$  with constructive parameter  $t$  is a one parameter set of points  $c(t)$ .

## 3.14. prism

<one parameter set of geometries> set of points in the union of the geometries (or the union of the trajectories) of a one parameter set of geometries [ISO 19141:2008, definition 4.1.18]

NOTE This is a generalization of the concept of a geometric prism that is the convex hull of two congruent polygons in 3D-space. Such polyhedrons can be viewed as a foliation of congruent polygons.

### **3.15. temporal coordinate system**

temporal reference system based on an interval scale on which distance is measured as a multiple of a single unit of time [ISO 19108:2002, definition 4.1.31]

### **3.16. temporal position**

location relative to a temporal reference system [ISO 19108:2002, definition 4.1.34]

### **3.17. temporal reference system**

reference system against which time is measured [ISO 19108:2002, definition 4.1.35]

### **3.18. trajectory**

path of a moving point described by a one parameter set of points [ISO 19141:2008, definition 4.1.22]

### **3.19. vector**

quantity having direction as well as magnitude [ISO 19123:2005, definition 4.1.43]

# Chapter 4. Conventions

This sections provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

## 4.1. Identifiers

The normative provisions in this document are denoted by the URI

<http://www.opengis.net/spec/{standard}/{m.n}>

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

### NOTE

A description of UML and how it is used in ISO and OGC standards will be included here.

# Chapter 5. Dynamic Features

Introductory text

## 5.1. Geometry in 3 Dimensions

**Topic:** Is the OGC Geometry model sufficient to support 3D?

### 5.1.1. Feature Model

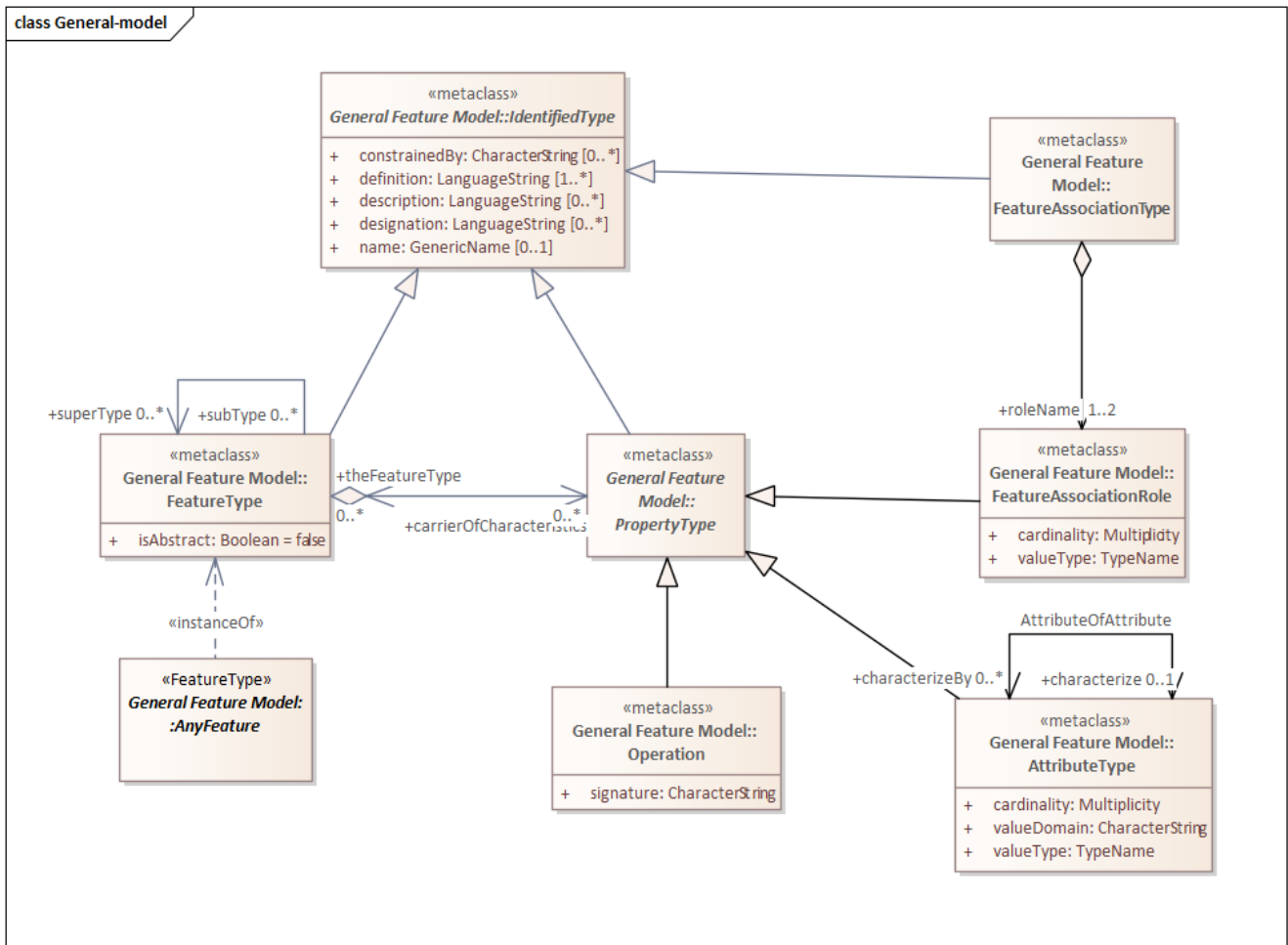
Before we can discuss geometry, we must first have a conceptual model of the universe. The OGC and ISO TC211 provide that model in ISO 19109. This standard defines the General Feature Model (GFM). The GFM defines the concept of a Feature, its components, and behaviors.

ISO TC211 defines a Feature as an "Abstraction of real world phenomena" (ISO 19101-1:2014)

A Feature, then, is a high level abstraction for anything that does or could exist in the universe.

The General Feature Model further refines the concept of a Feature. The following principles are most relevant for this discussion:

1. A Feature can be a FeatureType or an Instance of a FeatureType (AnyFeature).
2. FeatureTypes can form a taxonomy (inheritance)
3. Features possess characteristics (Properties)
4. A Property of a Feature can be an Operation, Attribute, or Association.



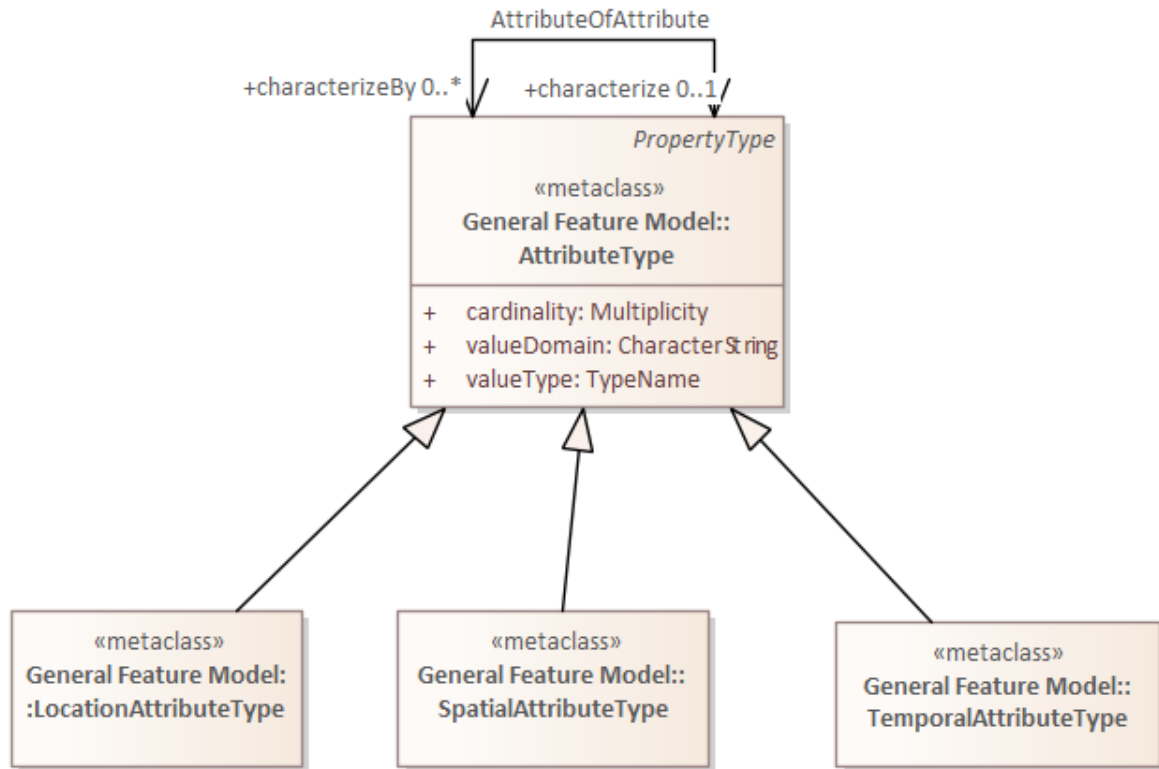
The resulting model is sufficient to describe a Feature's identity (IdentifiedType), what it is (FeatureType), what it can do (Operations), its observable characteristics (Attributes), and any associations with other Feature instances.

### 5.1.2. Feature Geometry

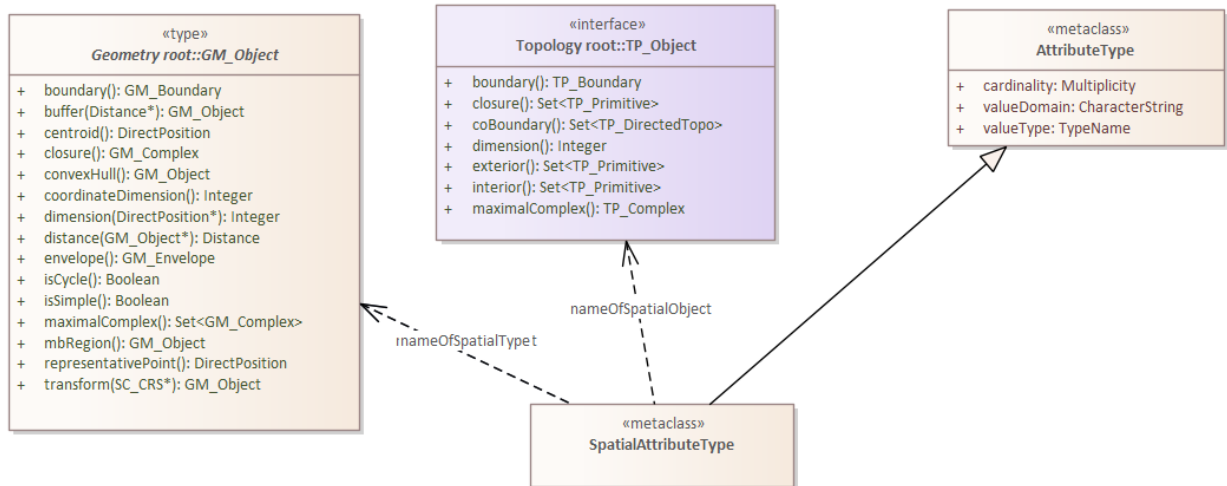
The General Feature Model treats geometry as an attribute of the Feature. In addition, it defines three types of attribute which are useful for associating geometry with a Feature in a standard manner:

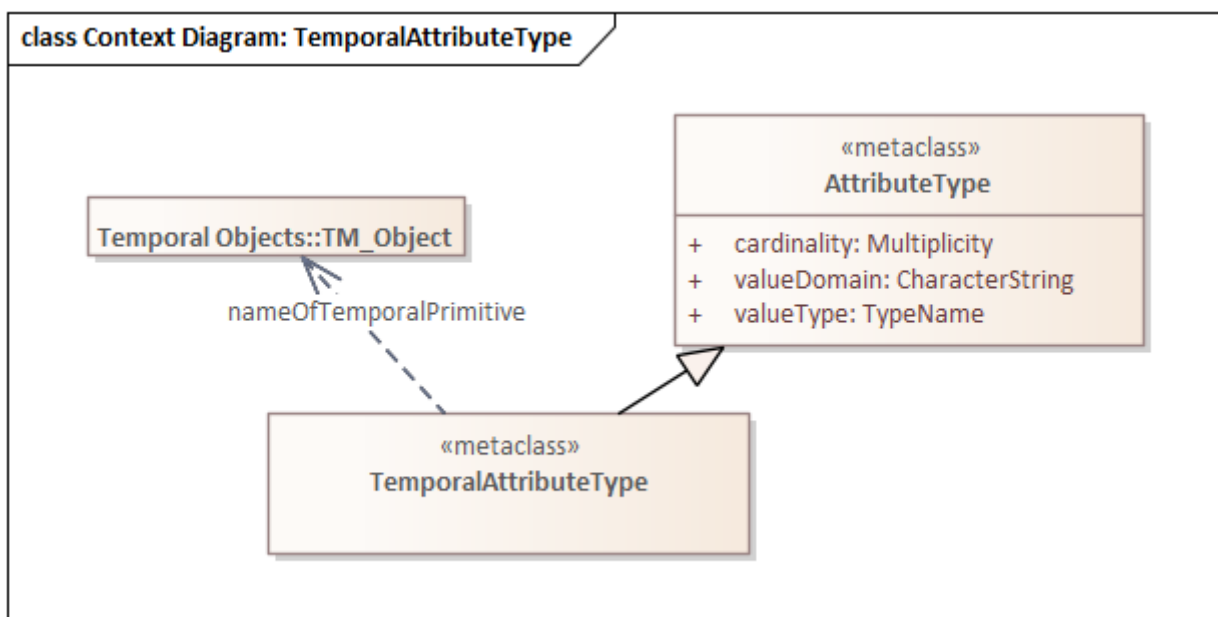
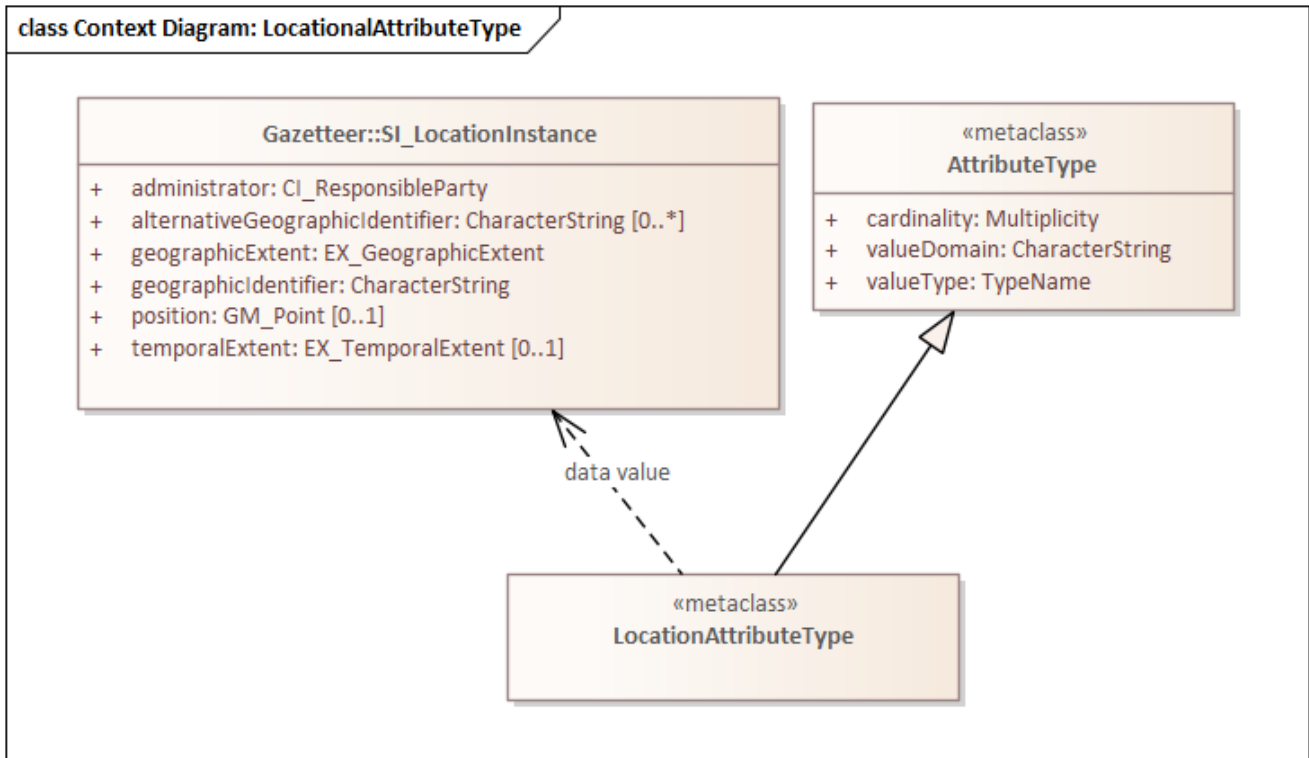
- SpatialAttributeType: Geometries (GM\_Object) and Topologies (TP\_Object)
- LocationalAttributeType: Named locations, extents, and points.
- TemporalAttributeType: Temporal objects (TM\_Object)

## class AttributeTypes



## class Context Diagram: SpatialAttributeType





In our discussion of Dynamic Features, we must allow for Features which are moving through three dimensions and have non-trivial three dimensional shapes. We must also consider that the shape of these objects may change with time. From this we see that the movement of the Feature and the shape of the Feature are two separate properties.

A measurement of movement would capture changes in location and orientation. Movement is in the eyes of an external observer. Therefore, movement should be specified using a coordinate reference system which is external to the Feature.

The shape of a Feature is independent of its location. A rigid body has the same shape regardless of where it is or who is observing it. It's geometry should be self-contained. This requires use of an internal coordinate reference system.

This leads up to two postulates:

Postulate 1: The Locational Attribute of a 3D Feature is a GM\_Point which locates the origin of the local CRS within an external CRS.

Postulate 2: The Spatial Attribute of a 3D Feature is one or more GM\_Objects which define the shape of the Feature in the local coordinate reference system.

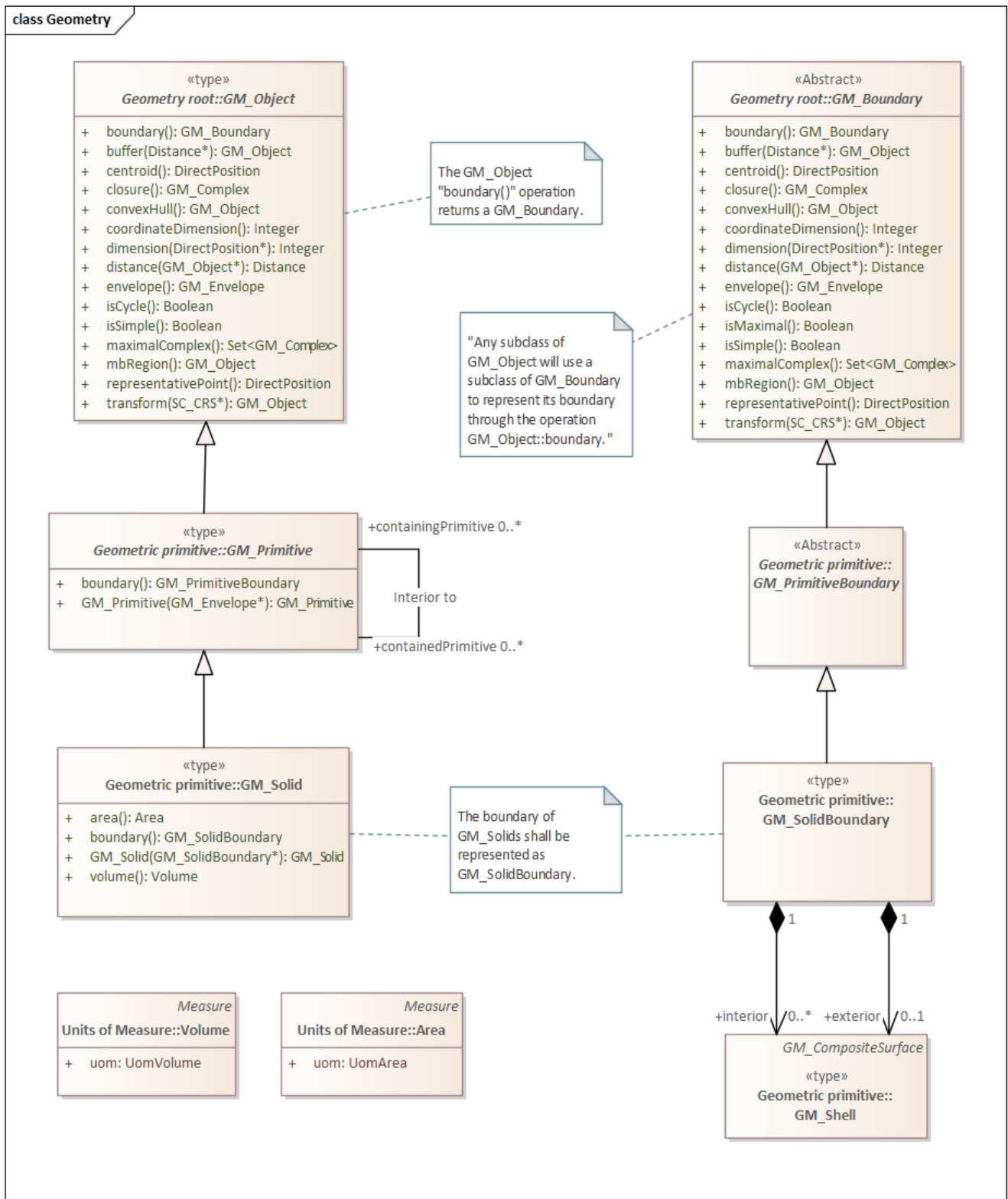
One and two dimensional geometries are well understood. So the locational attribute poses few immediate problems. Three-dimensional shapes are another matter. Particularly if they are free from any bounding surface and capable of movement.

### **5.1.3. 3D Geometry**

The applicable OGC/ISO standard for 3D geometries is ISO 19107:2003. While a new version was approved in 2019, it hasn't propagated through the rest of the ISO standards baseline. As a result, 19107:2003 is the most recent implemented version.

ISO 19107 defines both the GM\_Object and GM\_Boundary classes as root level geometry classes. The association between GM\_Object and GM\_Boundary is achieved through the "boundary()" operation on the GM\_Object class.





Therefore, we can describe the shape of a Feature in terms of the space occupied by that Feature and the boundary which encloses that space.

Postulate 3: A 3D Feature is a Feature whos' geometry is defined by a volume enclosed by a boundary surface.

## Volumetrics

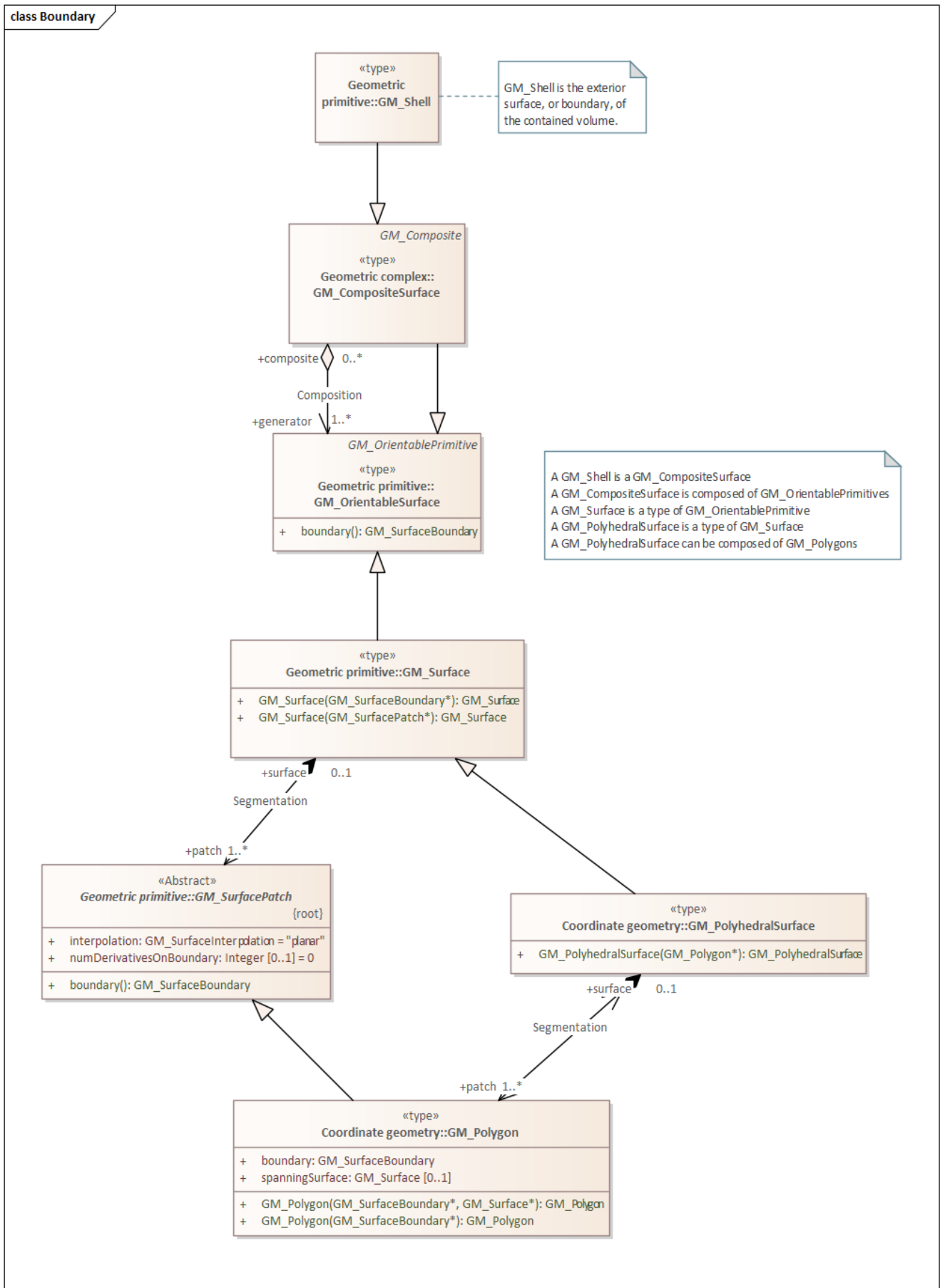
GM\_Object is subclassed into GM\_Primitive then into GM\_Solid. The "volume()" operation on GM\_Solid returns the volume (defined in ISO 19103) associated with a Feature. Thus, ISO 19107

supports the concept of a 3D volume.

Real 3D objects are often not solid. So the 3D model must also support voids, or even entire 3D Features within their interior. GM\_Primitive addresses this need through the “interior to” association. The two roles on this association are the containingPrimitive (the GM\_Primitive which contains another GM\_Primitive) and the containedPrimitive (the GM\_Primitive which is contained). This association has proven its worth in 2D space so there is little doubt that it will be just as effective in 3D.

## **Boundaries**

A 3D volume is delineated by a bounding surface. GM\_Boundary is the root class for boundaries. The subclass GM\_PrimitiveBoundary provides the boundary for GM\_Primitives. The GM\_PrimitiveBoundary subclass GM\_SolidBoundary is defined as the boundary for a GM\_Solid.



ISO 19107 goes even farther. A GM\_SolidBoundary is composed of both interior and exterior boundaries. These boundaries are defined by the GM\_Shell class. Following the class associations we see:

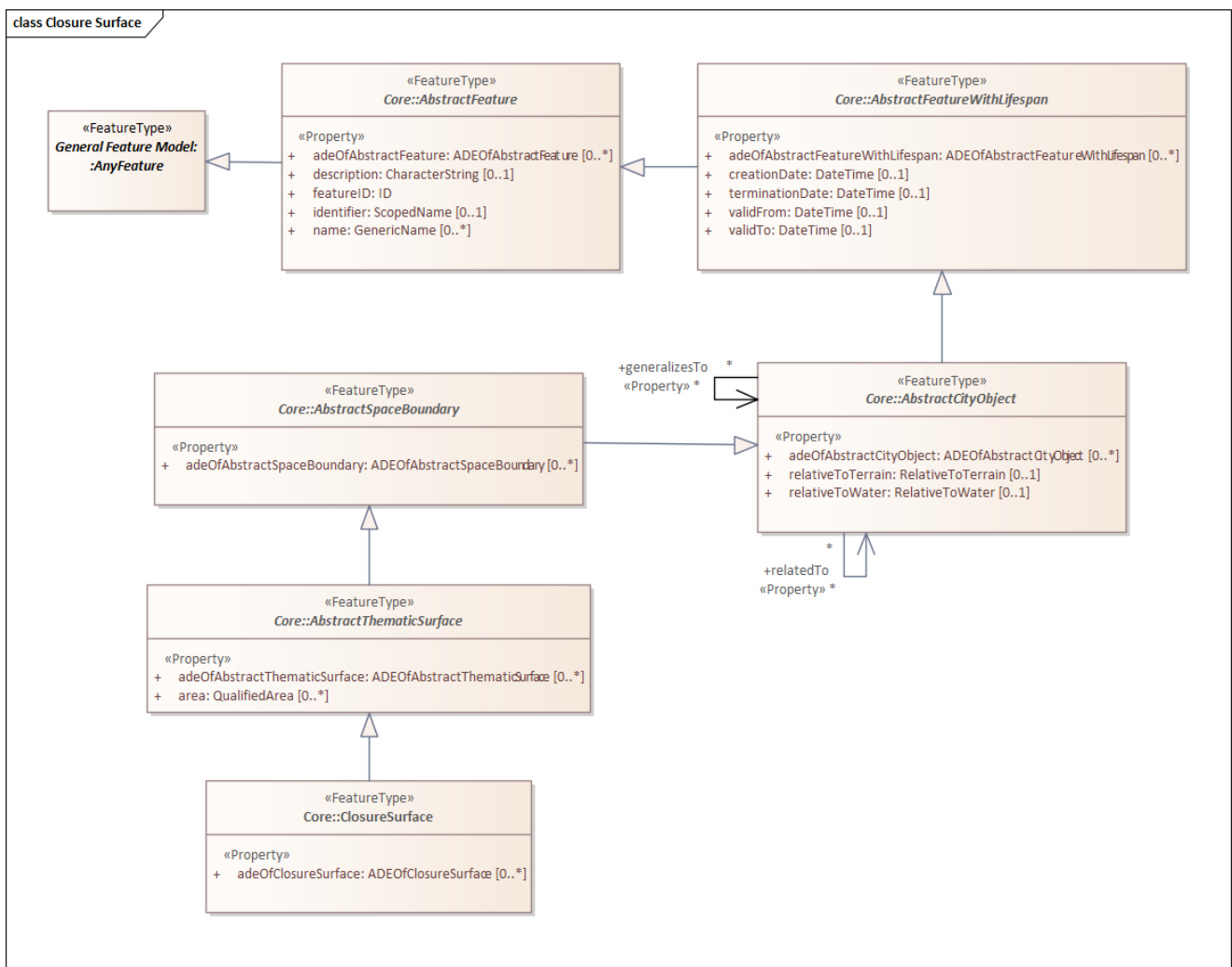
- A GM\_Shell is a GM\_CompositeSurface
- A GM\_CompositeSurface is composed of GM\_OrientablePrimitives
- A GM\_Surface is a type of GM\_OrientablePrimitive
- A GM\_PolyhedralSurface is a type of GM\_Surface
- A GM\_PolyhedralSurface can be composed of GM\_Polygons

A surface constructed of polygons is an example of Boundary Representation (B-Rep) of a surface. This approach is fundamental to rendering 3D computer graphics. (ref Adam Powers 1981)

## Closure Surfaces

Some structures, such as a tunnel or overpass, pose difficulties for this geometry model. The boundary surface can be constructed so that it continues into the interior of the structure. That would make the interior of a tunnel external to the tunnel object. This is not always a desirable result. CityGML provides the concept of a "Closure Surface".

A Closure Surface is a surface which is a logical part of the object but does not correspond to a physical part of the object. For example, the entrance to a tunnel can have a closure surface. This surface allows you to treat the tunnel as a single three-dimension entity, even though there is a hole in the bounding surface.



As implemented in CityGML 3.0, the ClosureSurface class has quite an ancestry. We may want to generalize this concept for use outside of CityGML. However, the capabilities provided by the ancestor classes do provide value and may be worth incorporating into a general 3D model.

### **B-Rep**

The polyhedral surfaces which bound volumetric shapes are similar to the Boundary Representation (B-Rep) approach used in CAD and computer graphics. B-Rep defines a 3-dimensional surface which serves as the interface between the interior of the volumetric shape and the exterior. This surface is usually defined by a collection of shape elements (polygons) which together form a closed surface.

[https://en.wikipedia.org/wiki/Boundary\\_representation](https://en.wikipedia.org/wiki/Boundary_representation)

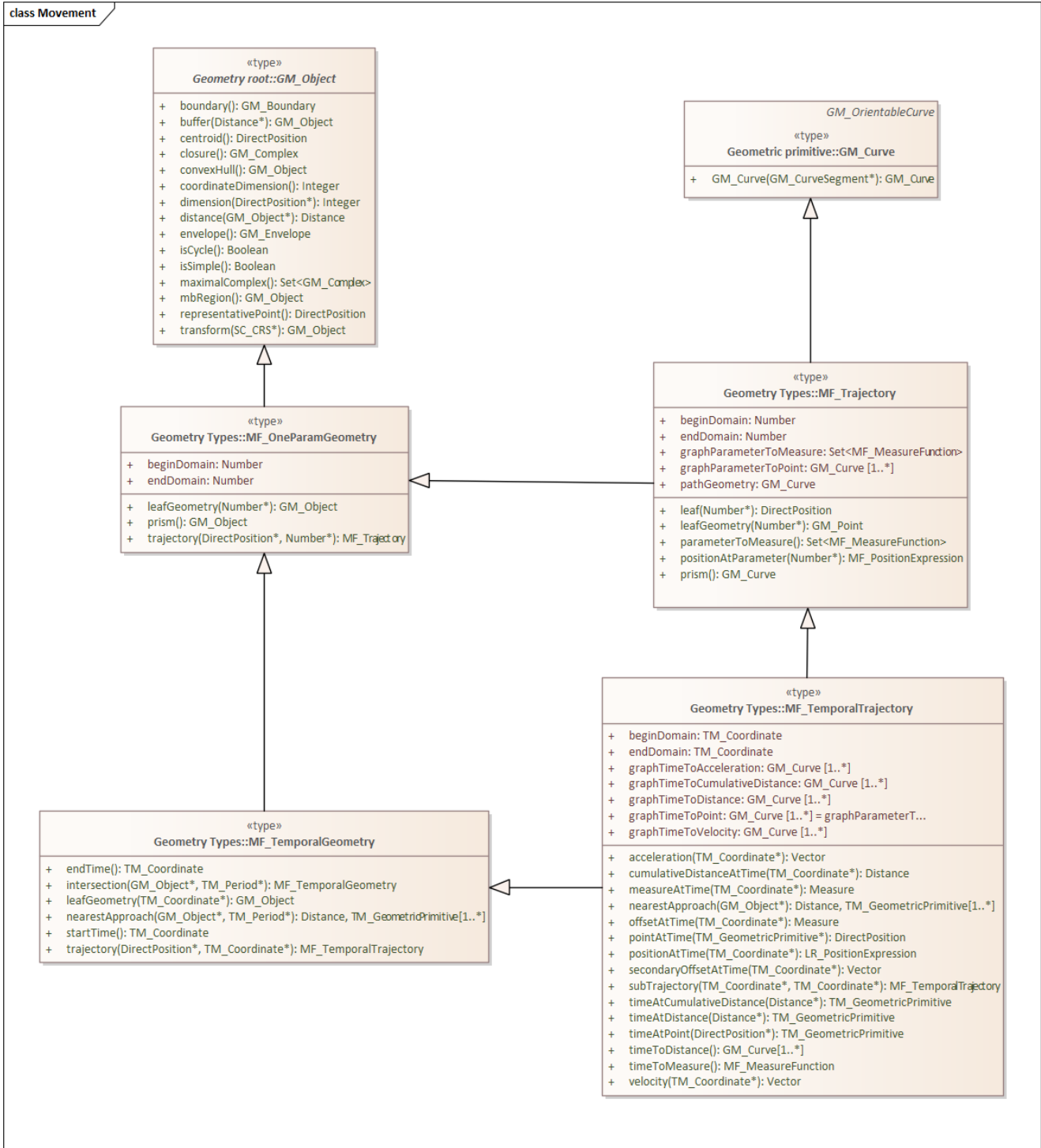
### **Point Clouds**

"In addition to the spatial representations defined in the Core module, the geometry of physical spaces and of thematic surfaces can now also be provided by 3D point clouds using MultiPoint geometry. This allows, for example, spatially representing the building hull, a room within a building or a single wall surface just by a point cloud. All thematic feature types including transportation objects, vegetation, city furniture, etc. can also be spatially represented by point clouds. In this way, the ClearanceSpace of a road or railway could, for instance, be modelled directly from the result of a mobile laser scanning campaign. Point clouds can either be included in a CityGML dataset or just reference an external file of some common types such as LAS or LAZ."

## **5.2. Moving Features**

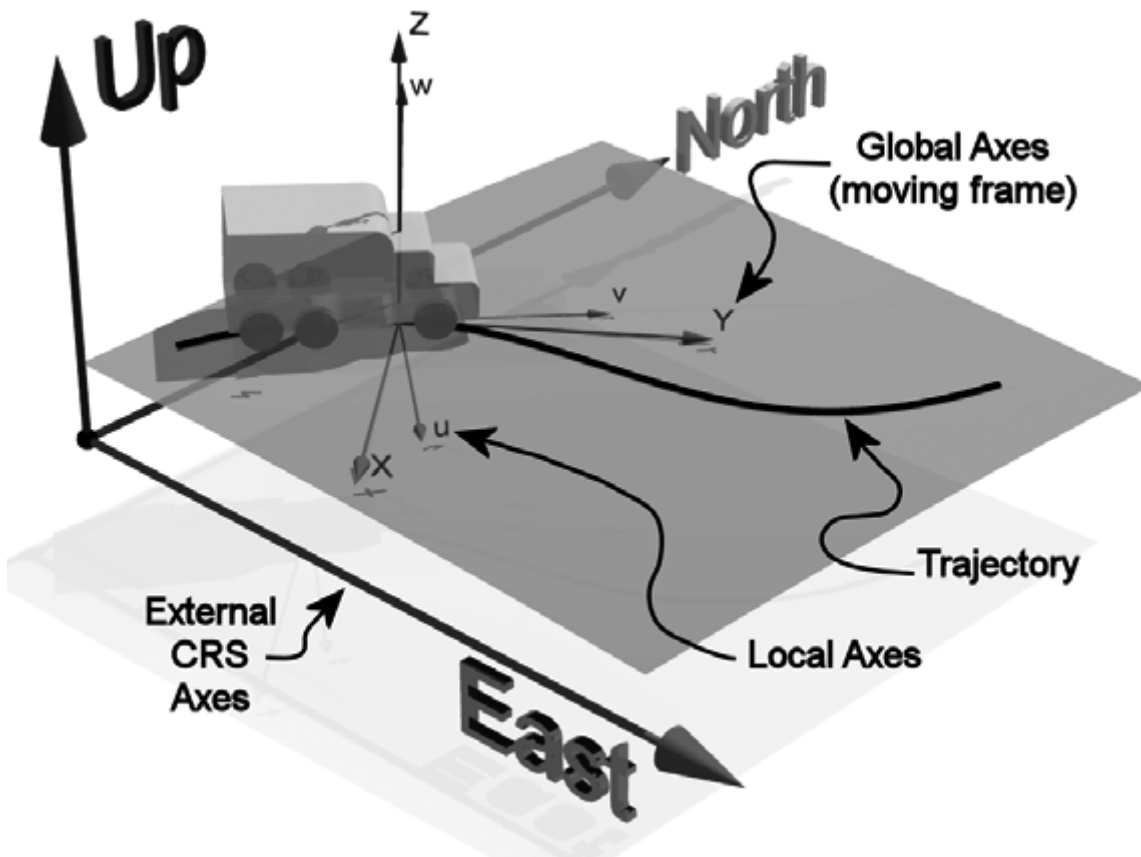
**Topic:** Is the OGC geometry model sufficient to support 3D Moving Features?

The ISO Standard for Moving Features is ISO 19141:2008. This standard extends the geometry model from ISO 19107 and, by association, the Feature Model from ISO 19109.



A high level view of ISO 19141 is provided in Figure --. The classes identified in this figure are described below. But first, a discussion of coordinate reference systems (CRS) is in-order.

### 5.2.1. Coordinate systems



Moving Features deal with three spatial coordinate systems as well as one temporal coordinate system. The spatial coordinate systems are referred to as the External, Global, and Local CRS.

When dealing with Moving Features, we frequently need to convert coordinates between the three spatial CRS. The `GM_Object` class provides us with the `transform()` operation which can be used for this purpose.

### External CRS

The External coordinate system is the coordinate system within which the Moving Feature exists. Typically this is an Earth-centric geographic CRS.

### Global CRS

The Global coordinate system is a moving CRS which describes the area external to the Feature at a specific time. Typically the origin is the location of the Feature on the trajectory and the principal axis is the tangent to the trajectory at that point.

### Local CRS

The local coordinate system is internal to the Feature. This is usually a cartesian coordinate system with the origin at a prominent point in the Feature such as the center of mass. The origin of the local coordinate system should also be the point where the location and orientation of the Feature is measured.

## Temporal Reference Systems

ISO 19108:2006 Geographic Schema - Temporal Schema is the ISO standard for Temporal Reference Systems. In particular, the `TM_ReferenceSystem` class.

`TM_ReferenceSystem` has two attributes; `domainOfValidity` and `name`. The `name` attribute is an identifier for this temporal reference system. The `domainOfValidity` specifies the spatial extent over which this TRS is applicable.

`TM_ReferenceSystem` is specialized through a number of subclasses. The two most relevant to this paper are `TM_CoordinateSystem` and `TM_Clock`.

`TM_CoordinateSystem` is "A system for measuring time on a continuous interval scale using a single standard time interval". The standard time interval is provided through the `interval` attribute. In addition, the `origin` attribute provides a temporal "datum" from which time is measured. Since time is a one-dimensional quantity, the origin and interval are sufficient to define a basic Temporal Coordinate Reference System.

`TM_Clock` is "A system for measuring temporal position within a day". It has an optional `dateBasis` association with a calendar (`TM_Calendar`).

This combination of classes allows us support high precision local-clock TRS as well as full date-time TRS.

### 5.2.2. Coordinate Representation

The coordinates used to define a 3D moving geometry (MG) face requirements specific to their use. These requirements are derived from two characteristics of moving geometries. Unlike static spatial geometries, time and location in moving geometries are tightly coupled. They must act as a single, four dimension location. In addition, there will be a large number of coordinate measurements. This is a result of the need to accurately track movement over time.

1. A MF coordinate must represent a discrete location in space and time.
2. A MF coordinate must include values for all three spatial axis (X,Y,Z) as well as the temporal axis (t).
3. A MF coordinate must be concise.

Of the Moving Feature encoding standards, the JSON encoding comes closest to meeting these requirements. Its major shortfall is the need for conformance with GeoJSON. Since GeoJSON assumes a terrestrial spatial geometry, spatial and temporal coordinates must be encoded separately.

- A `LinearTrajectory` object SHALL be a GeoJSON Feature object that has two MANDATORY members of "geometry" and "properties".

The spatial locations are captured using the GeoJSON "geometry" property. This property is restricted as follows:

- The value of the "geometry" member SHALL be a `LineString` Geometry object, having "type" = "LineString".



- The number of elements in the array of the "coordinates" value in the Geometry object SHALL more than two positions.

So the spatial geometry is a linestring of more than two points.

GeoJSON does not support temporal coordinates directly. So the "properties" property is adapted for this purpose. Since "properties" is not limited to temporal coordinates, these requirements are more complex.

- The value of the "properties" member SHALL be a GeoJSON object that has at least a member with the named "datetimes".
- The value of the "datetimes" member is a JSON array.
- Each element in the array of the "datetimes" value SHALL be an instant object.
- An instant object SHALL be only a JSON string to represent a timestamp encoded by the IETF RFC 3339 format using **Z** or the numeric value of milliseconds since midnight (00:00 a.m.) on January 1, 1970, in UTC.
- The array of the "datetimes" value SHALL be a monotonic increasing sequence.
- There SHALL be no instant object that has the same value as any other element.

The consequence of these requirements is that the "properties" property can carry the temporal equivalent to a line string. There is one final requirement.

- The number of elements in both arrays of the "coordinates" value and the "datetimes" value SHALL be equal.

So there is a one-to-one correspondance between the temporal measurements in the "properties" property and the spatial measurements in the "geometry" property.

An example of this encoding is provided ---

```
{
  "type": "Feature",
  "id": "A",
  "geometry": {
    "type": "LineString",
    "coordinates": [[11.0,2.0,50.0], [12.0,3.0,52.0], [10.0,3.0,56.0]]
  },
  "properties": {
    "datetimes": ["2012-01-17T12:33:51Z", "2012-01-17T12:33:56Z", "2012-01-17T12:34:00Z"],
    "state": ["walking", "walking"],
    "typecode": [1, 2]
  }
},
```

### 5.2.3. Geometry

#### MF\_OneParameterGeometry

We start our discussion with the class MF\_OneParameterGeometry. MF\_OneParameterGeometry is a subclass of GM\_Object. So moving features have the 3D geometric properties of any other GM\_Object. What is different is that this geometry can change as a function of a parameter.

A one parameter set of geometries is defined as "a function  $f$  from an interval  $t \in [a, b]$  such that  $f(t)$  is a geometry and for each point  $P \in f(a)$  there is a one parameter set of points (called the trajectory of  $P$ )  $P(t) : [a, b] \rightarrow P(t)$  such that  $P(t) \in f(t)$ . A leaf of a one parameter set of geometries is the geometry  $f(t)$  at a particular value of the parameter".

A one parameter geometry instance includes a "leafgeometry()" operation. This operation takes the parameter ( $t$ ) as input and returns the leaf  $P(t)$  for that parameter as a GM\_Object.

#### MF\_TemporalGeometry

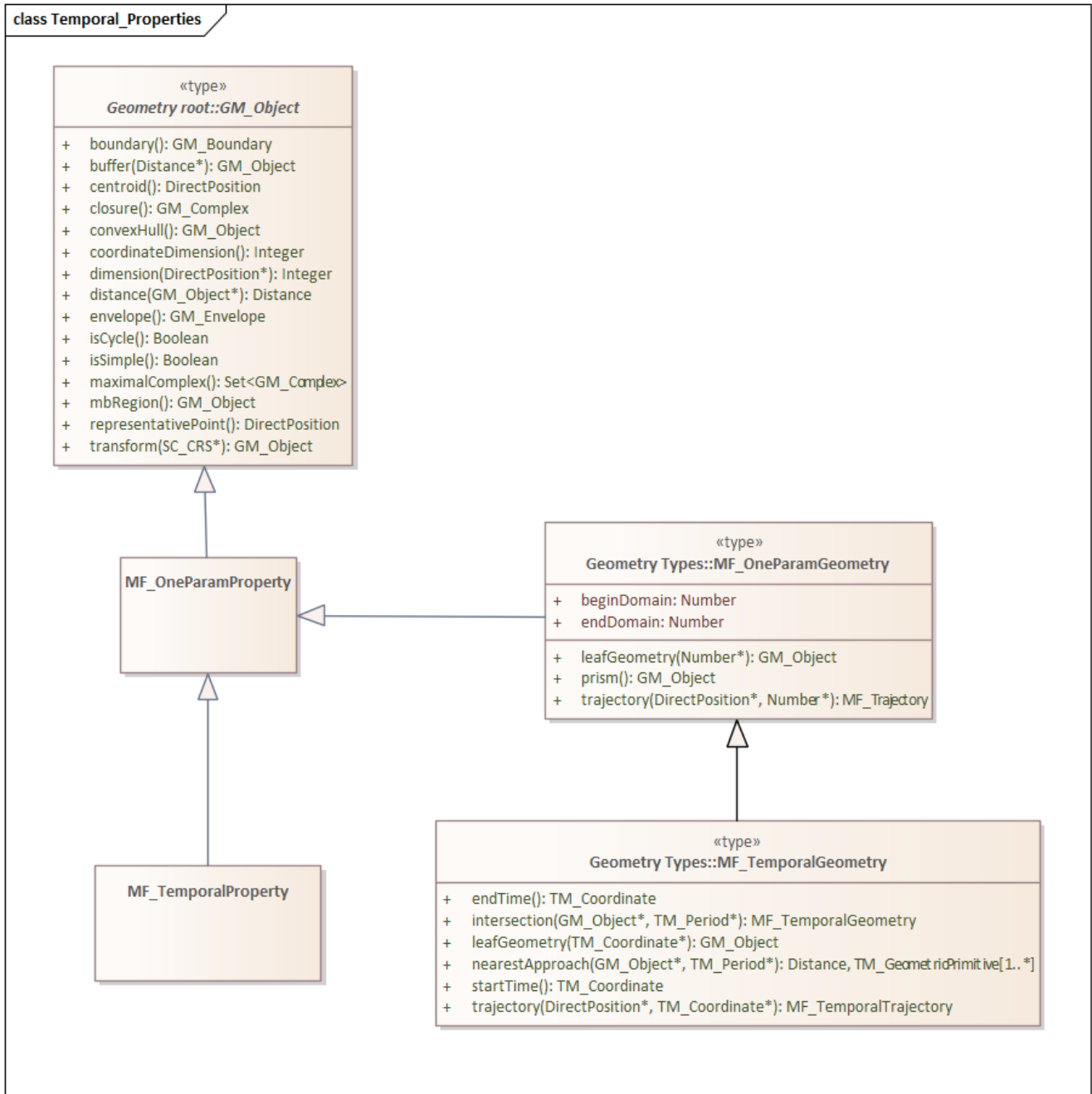
An MF\_TemporalGeometry is a MF\_OneParameterGeometry where the parameter is Time expressed as a TM\_Coordinate. TM\_Coordinate is specified in ISO 19108; it expresses time as a multiple of a single unit of measure such as year, day, or second. The "leafgeometry()" operation of an instance of MF\_TemporalGeometry would take a TM\_Coordinate in as input and return a GM\_Object instance representing the geometry of the Feature at the specified point in time.

#### Temporal Properties

The JSON encoding of the OGC Moving Features standard introduces the concept of temporal properties.

"A TemporalProperties object is a JSON array of ParametricValues objects that groups a collection of dynamic non-spatial attributes and its parametric values with time."

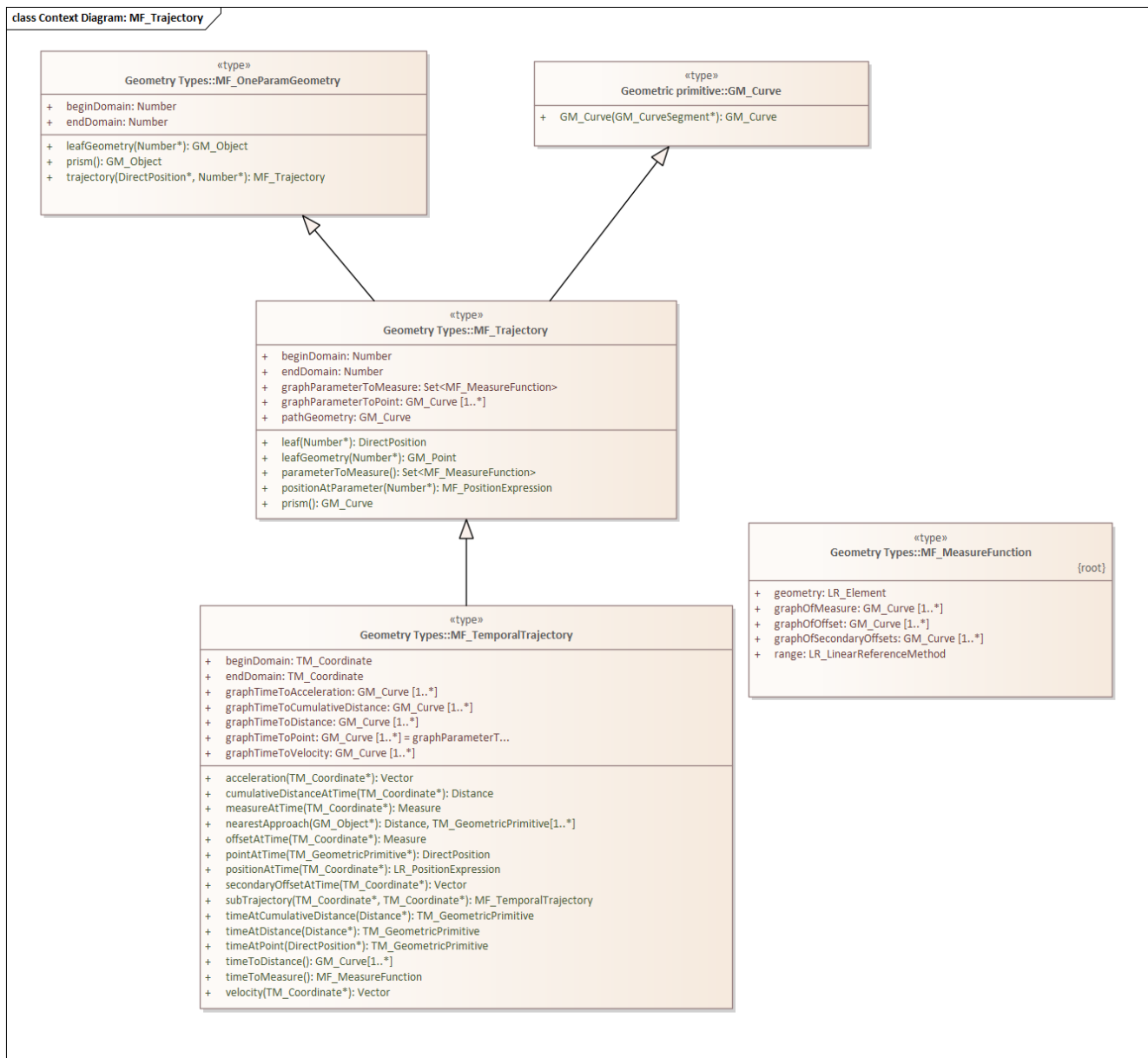
Logically TemporalProperties should be a subclass of OneParamProperties. Since Geometry is a property, then MF\_TemporalGeometry should be a subclass of TemporalProperties. Which gives us the following UML.



Temporal properties are particularly useful for capturing state change. For example, the fuel load of an aircraft will change over time. The `leafproperty()` operation on a temporal `fuel_load` object would return the amount of fuel onboard at the specified time.

#### 5.2.4. Location

ISO 19141 represents the location of a Moving Feature using two classes; `MF_Trajectory` and `MF_TemporalTrajectory`.



A MF\_Trajectory is a curve (GM\_Curve). It represents every position that the Feature has occupied during its journey. It does not necessarily represent the time when each location was reached.

MF\_TemporalTrajectory makes the MF\_Trajectory a MF\_TemporalGeometry. It represents location along the trajectory as a function of time. So each location is fully defined in both space and time.

A Temporal Trajectory has two operations of particular interest; `leaf()` and `leafgeometry()`. The input parameter for these operations is always time (TM\_Coordinate).

The `leaf()` operation returns the spatial location (Direct\_Position) that the Moving Feature passes at the time (TM\_Coordinate) specified by the input parameter. This is a point on the trajectory GM\_Curve geometry. It also serves as the origin of the Global CRS at that location on the trajectory.

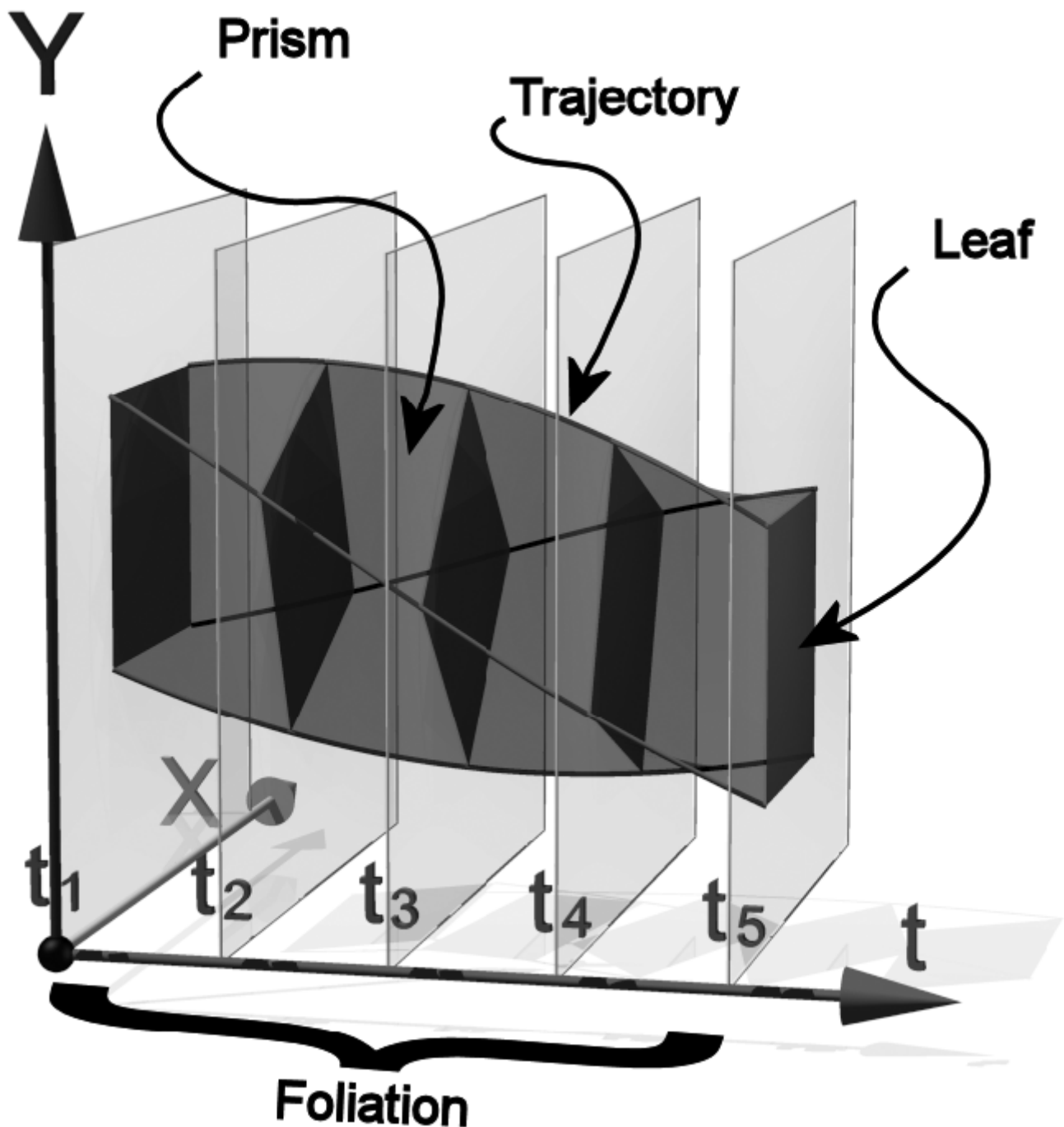
The `LeafGeometry()` operation returns the spatial geometry (GM\_Point) that is Moving Feature passes at the time (TM\_Coordinate) specified by the input parameter. This is the geometry of the Moving Feature expressed in the Local CRS. Since Trajectories only convey location, only GM\_Point geometries are supported.

## 5.2.5. Orientation

### MF\_PrismGeometry

If an application focuses on only the linear movement (i.e., the spatiotemporal line string) of moving points based on World Geodetic System 1984, with longitude and latitude units of decimal degrees, and the ISO 8601 standard for representation of dates and times using the Gregorian calendar, the application can share the trajectory data by using **only** IETF GeoJSON, called **MF-JSON Trajectory**. For other cases, **MF-JSON Prism** can be used for expressing more complex movements of moving features. **MF-JSON Prism** is a GeoJSON-like format reserving new members of JSON objects ("temporalGeometry," "temporalProperties," "crs," "trs," "time," and others) as "foreign members" to represent spatiotemporal geometries, variations of measure, coordinate reference systems, and the particular period of moving features in a JSON document.

A trajectory provides the location of a Moving Feature as a function of time. Prism Geometry represents the full geometry (location, orientation, and shape) of the Feature as a function of time.



The key concepts in the Prism model are:

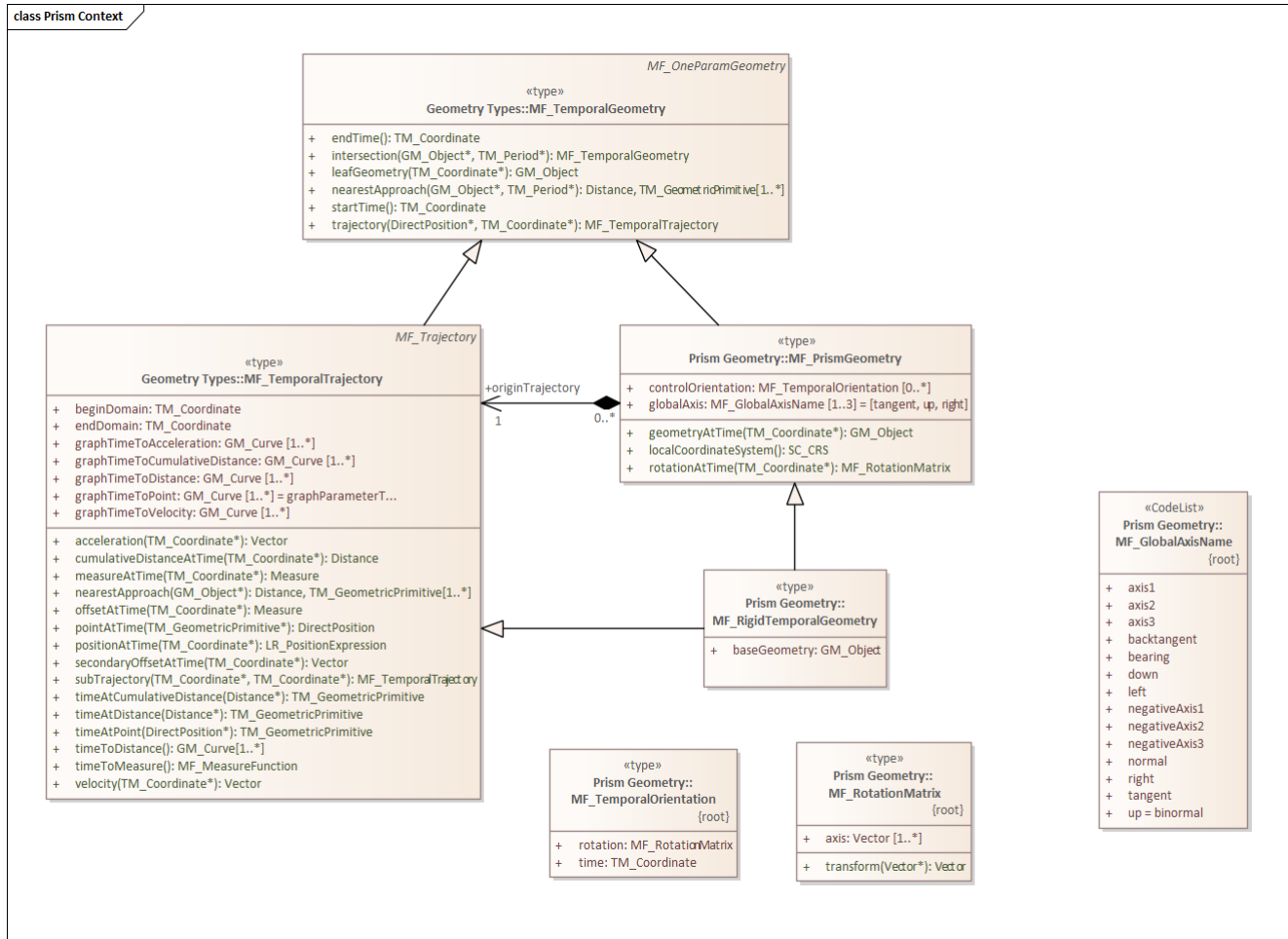
**Leaf:** A leaf is the geometry of the Moving Feature at time ( $t_n$ ).

**Foliation:** A collection of leaves where there is a complete and separate representation of the geometry of the Feature for each specific time ( $t_n$ ).

**Trajectory:** A curve that represents the path of a point in the geometry of the Moving Feature as it moves with respect to time ( $t$ ).

**Prism:** the union of the geometries (or the union of the trajectories) in a foliation.

Like a Temporal Trajectory, a Prism is a subclass of MF\_TemporalGeometry.



A MF\_PrismGeometry class has the following characteristics.

The association role "originTrajectory" associates a Temporal Trajectory with a Prism. For any TM\_Position, the associated Temporal Trajectory provides the location of the Moving Feature in the Global CRS. This point serves as the origin of the Local CRS.

The localCoordinateSystem() operation returns a SC\_CRS for the design coordinate reference system in which the moving feature's shape is defined.

The rotationAtTime() operation accepts a time in the domain of the prism geometry and returns the rotation matrix that embeds the local geometry into geographic space at a given time (TM\_Coordinate). The vectors of the rotation matrix allow the feature to be aligned and scaled as appropriate to the vectors of the global coordinate reference system.

This one association and two operations provide us with the location, orientation, axis definition, and units of measure needed to define identify the local CRS and to transform geometries between the Local and Global CRS.

Finally, the geometryAtTime() operation accepts a time in the domain of the prism geometry and returns the geometry of the moving feature, as it is at a given time in the global coordinate reference system. The return type is a GM\_Object so this operation is not limited to points. It is fully capable of representing a 3D surface and volume.

In short, a MF\_PrismGeometry provides us with the shape, location, and orientation of a Moving Feature as a function of time (tn).

### 5.2.6. Non-rigid Bodies

ISO 19141 only addresses rigid bodies. The shape returned by a `geometryAtTime()` operation will always be the same. However, it leaves open the opportunity to extend the Moving Feature model to support plastic (non-rigid) objects.

The most obvious approach is to allow the geometry returned by the `geometryAtTime()` operation to change as a function of time. This doesn't require a change to the model. But it may require some changes to the standard.

As a correlary to this approach, the geometry itself could include `MF_TemporalGeometry` elements. These elements would each have their own lifespan. A history of their movement, in respect to the local CRS, over time.

## 5.3. Articulated Geometries

Given a suite of standards which allow you to define time-variant geometric elements, then the next step is to take a collection of those elements and assemble them into a complex object.

An articulated geometry is such an aggregation where each element has less than 6 degrees of freedom. Each element can move, but its movement is constrained by attachment to one or more additional elements.

The aggregate as a whole can also move, but that movement becomes more complex. typically we would model movement of the whole as a trajectory of the center of mass. However, the center of mass of an articulating Feature will change as the relative position of the elements change.

### 5.3.1. GeoPOSE

The Frame Transform is a representation of the transformation taking an Outer Frame coordinate system to an Inner Frame coordinate system. This abstraction is constrained in GeoPose v 1.0 to only allow transformations involving translation and rotation. The intention is to match the usual concept of a pose as a position and orientation. The formalism that expresses a GeoPose Frame Transform is a pair of Reference Frames, Outer and Inner, each defined by a Frame Specification.

A pose has a valid time (`GeoPose_instant`)

Discuss outer frame vs Inner Frame

EPSg 4979 is basis for all frames?

insert figure 8

Sequence

The sequence logical model defines a method for packaging of GeoPose data, where multiple GeoPoses in a sequence share the same Outer Frame and there is a time-dependent changing Inner Frame.

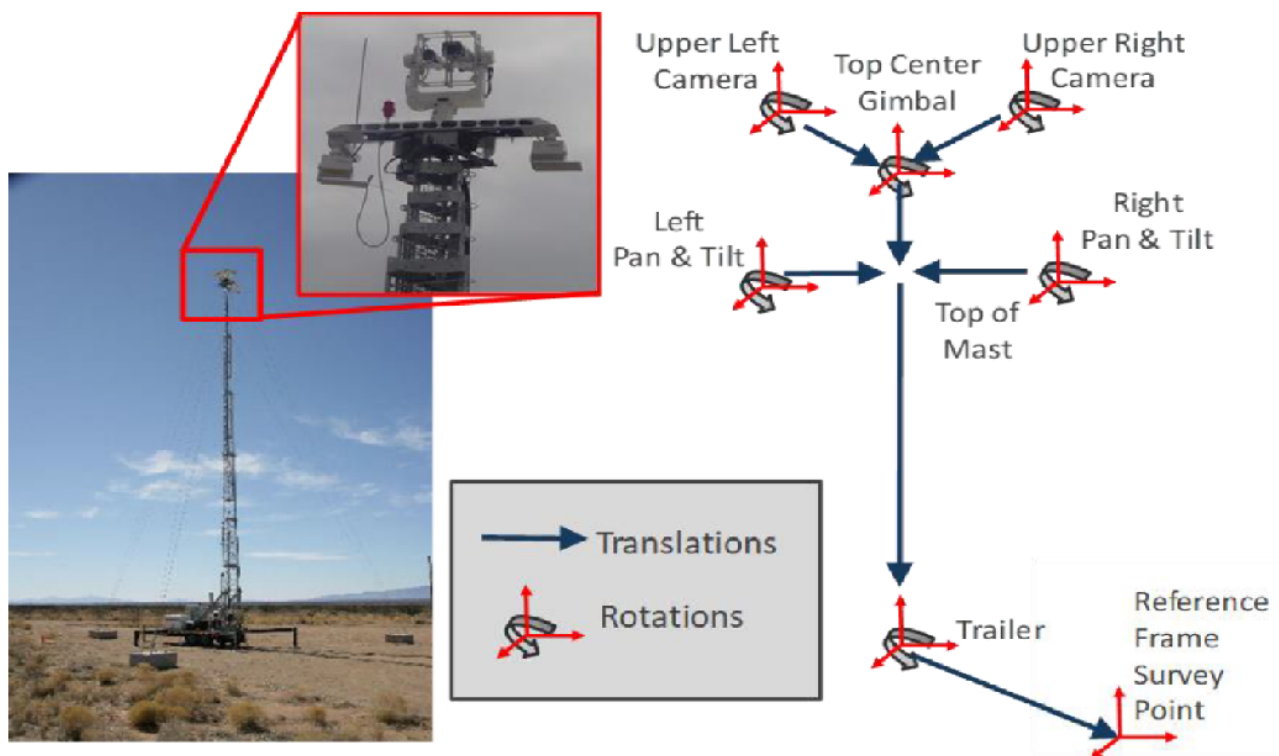


### 5.3.2. MISB Staging System

#### MISB 1906 Motion Imagery Metadata (MIMD): Staging System

"The Motion Imagery Metadata (MIMD) Model includes locations, orientations, and kinematics (velocity, acceleration, etc.) of platforms, gimbals, sensors, sensor elements, geospatial and other points. The locations and orientations are either absolute references to a well-known frame of reference (e.g., WGS84) or relative references to other locations and orientations. Each location and orientation pairing define a "stage" that has the potential to be the frame of reference for another location and orientation. Linking stages together forms the Staging System. The Staging System then defines an ability to describe, in metadata, the physical make-up and configuration of a system and the time varying physical relationships of the system and its sub-system components."

The MISB Staging System was designed to capture the instantaneous configuration of an articulated sensor system.



Example Motion Imagery System with Multiple Sensors and Gimbals (Photo credit White Sands Missile Range)

Stage: a single frame of reference located at a point. It defines the location, orientation, and kinematics of a coordinate system located at that point. These properties can be defined in terms of absolute values, or as relative values measured from an "parent" reference system.

Constellation: A system of one or more stages where the parent-child relationships between the stages is sufficient to calculate the absolute values for every stage.

Root Stage: This is the starting point for a Constellation. This stage is expressed in absolute values.

Table 1. Stage System Example

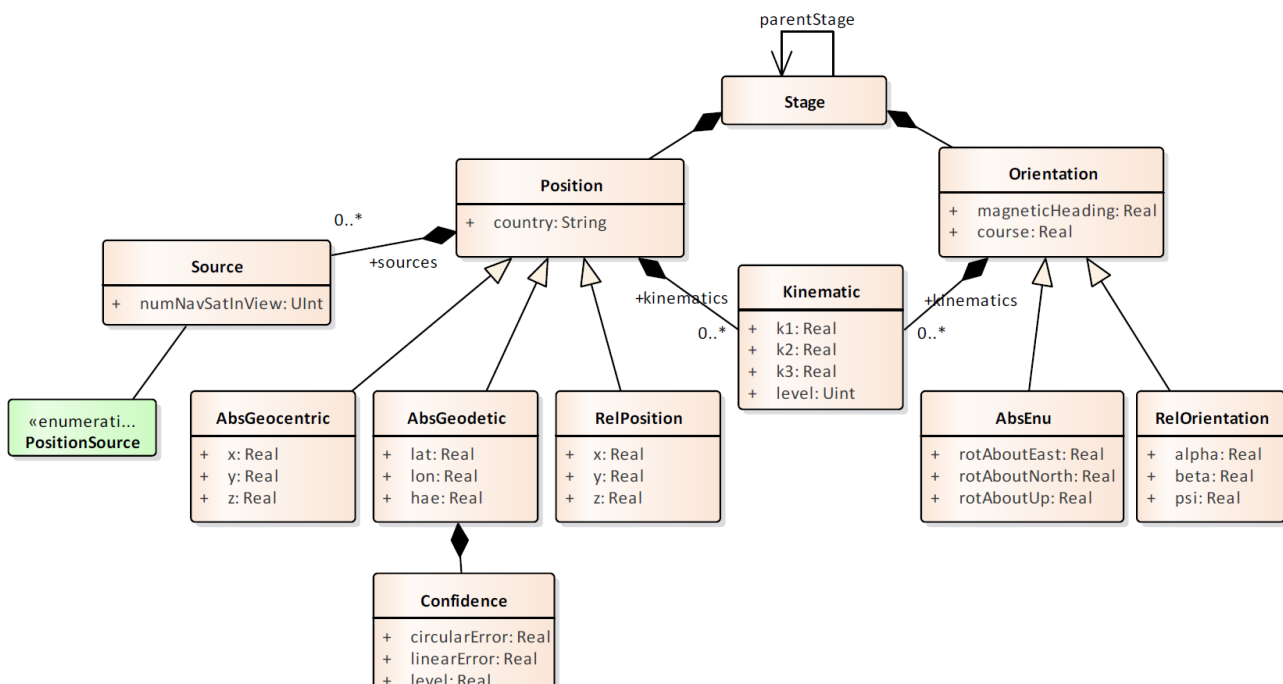
Stage	Component	Parent	Values
1	Reference Frame Survey Point	0	Absolute Position/Orientation
2	Trailer	1	Relative Position / Orientation
3	Left Camera	2	Relative Position / Orientation
4	Right Camera	2	Relative Position / Orientation
5	Top Center Gimbal	2	Relative Position / Orientation
6	Left Upper Camera	5	Relative Position / Orientation
7	Right Upper Camera	5	Relative Position / Orientation

The Staging System's absolute positions use either WGS84 Ellipsoid angular coordinates (i.e., geodetic Latitude, Longitude, Height above Ellipsoid (HAE)), or WGS84 geocentric Earth-Centered Earth-Fixed (ECEF) Cartesian coordinates (i.e., X, Y, Z).

The Staging System's relative positions use Cartesian coordinates (i.e., x, y, z) measured in meters from the parent stage frame of reference.

The Staging System's orientations use Euler rotations, measured in radians, around three axes (X, Y, and Z) of a righthanded coordinate system. The Staging System's absolute orientation has the X, Y, and Z axes aligned with the East, North, and Up (ENU) respectively.

The Staging System standardizes rotations to use a specific order of rotations. The Staging System uses Tait-Bryan angles with a Primary Rotation Order of Z-Y-X (see Appendix C); thereby the first rotation is around the Z-axis, the second rotation is around the new Y-axis (after rotation around the Z-axis), the final rotation is around the new X-axis (after rotation around the Z and then Y axes).



The Staging system does not explicitly address temporality. However, this information is passed in the MISB KLV metadata stream. As such, each update is associated with the precision time stamp of that packet. In addition, an update can be associated with a Timer as described in section ---

### 5.3.3. Discussions

Quaternions vs. Euler rotations

Temporal representation

Standardize on a right-handed Cartesian coordinate system for all stages/Poses?

Common concepts:

1. An "anchor" node which ties the relative coordinates to an absolute CRS
2. Local coordinate systems associated with each other through standardized transformations

## 5.4. Mass Properties

Consider an articulated vehicle in free-flight. As the components change position, they change the center of mass, principle axis of rotation, and a number of other mass properties. These changes affect the dynamic behavior of the vehicle. Most designers take great care to assure that problems cannot occur. But that doesn't mean we should ignore the issue.

Comments?

## 5.5. Temporal Reference Systems

Proposition: Time is in the eye of the beholder. So all measurements of time must be local.

Dynamic Features are not tied to an Earth-centered static existence. Yet the concepts of time used in the geospatial community are almost exclusively based on Earth-centric astronomical phenomena. They also assume a rather coarse degree of granularity. For dynamic features we need to use local clocks with precision down to the nanosecond. We are less concerned with absolute time than with relative time. State B was achieved 37 nanoseconds after State A.

It's only when we begin aggregating these dynamic elements that we begin to worry about "absolute" time. Even then, we are more likely to convert from one local clock to another then to convert to an absolute time.

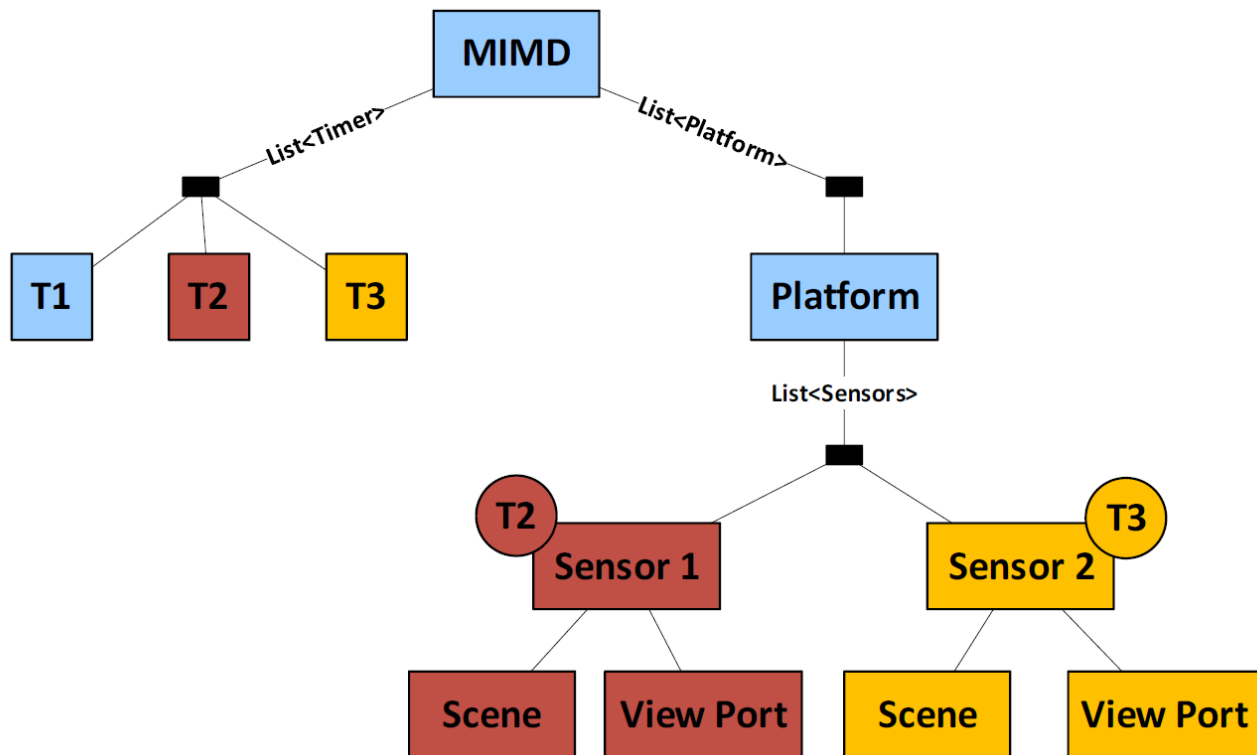
So if all time is local, we need a Temporal Reference System concept which captures the parameters needed to transform across TRSs. A temporal equivalent to GeoPOSE.

### 5.5.1. MISB Timers

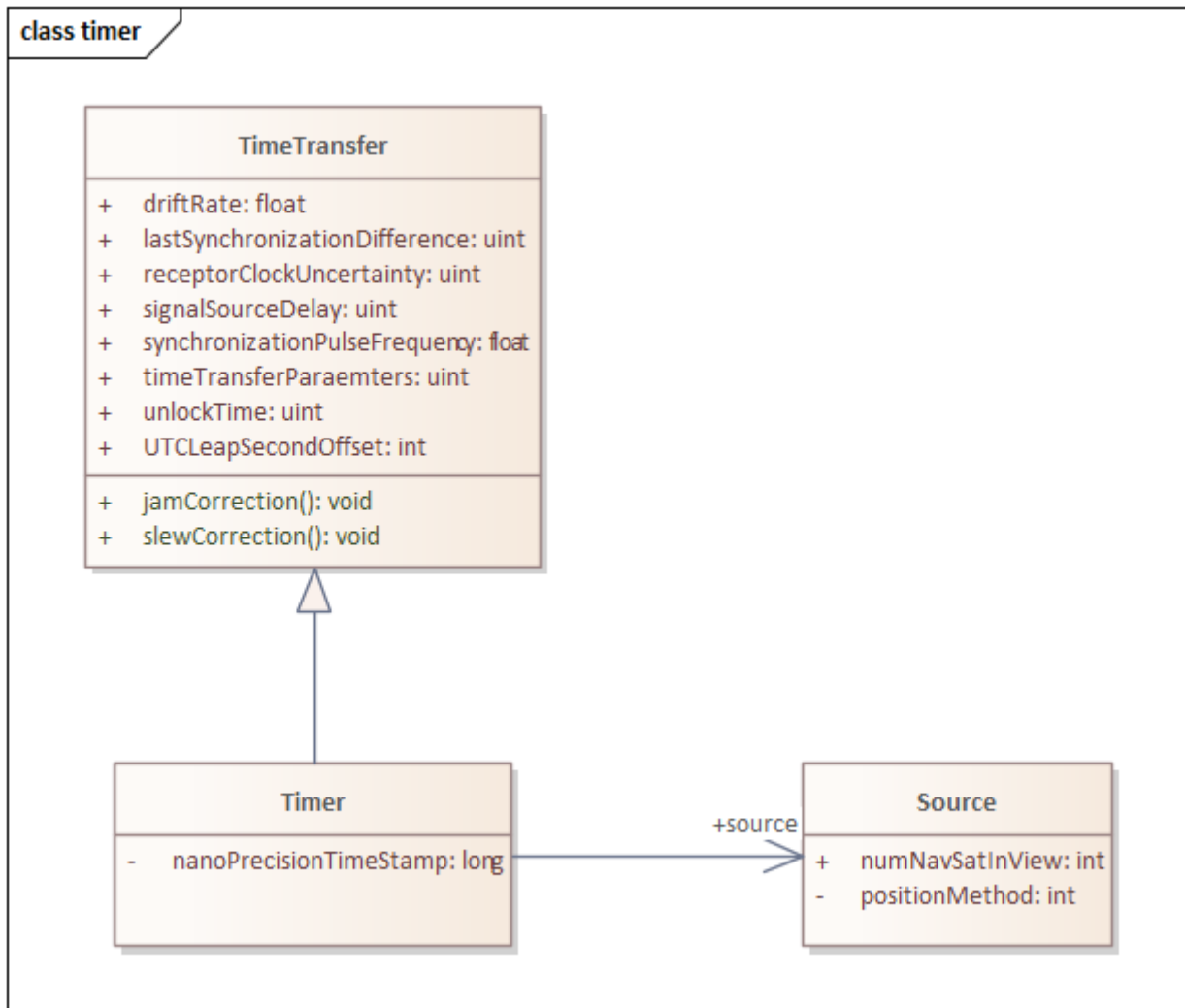
Timing is critical for motion imagery. Not only are accurate time stamps essential for the smooth viewing of the images, they are even more critical for the accuracy of data derived from that imagery. Proper geolocation of a moving image, for example, requires nano-precision time stamps.

For that reason, the Motion Imagery Standards Board (MISB) developed a timing infrastructure for use with MISB standards. This infrastructure addresses not only the locality of time, but also the conversion of time values from one locality to another.

Three timers are illustrated in the example below (T1, T2, and T3). T1 is associated with the Platform. This timer receives regular GPS updates and is considered the most accurate in this system. T2 is associated with Sensor 1. This is a simple oscillator, counting nanoseconds since the sensor was last powered up. T3 is associated with Sensor 2. It also is a simple oscillator.



While timers T2 and T3 are sufficient for the operation of their associated sensors, data based on time values from these Timers would not be usable. A transformation from the local time to global time is required.



Discussion of Timers and how their attributes are used to build a mesh of TRS - similar to GeoPOSE.

### 5.5.2. Temporal Schema

The ISO standard covering temporal reference systems is ISO 19108:2002 Geographic Information - Temporal Schema.

Discuss Calendars and clocks

Local vs "Global" clocks

POSIX time is tied to a global datum, a calendar.

Local time is tied to a local counter.

Analogous to a Spatial Reference System vs. an Engineering CRS.

Extending TM\_Clock and/or TM\_coordinateSystem

## 5.6. Dynamic Features at Relativistic Velocities

At very high velocities, measurements between moving features suffer from relativistic effects. The proposition is to build on Minkowski SpaceTime to address these cases.

### 5.6.1. Lorentz Transformation

A transformation between two different coordinate frames that move at a constant velocity and are relative to each other.

ONLY related to changes in inertial reference frames:

- Inertial frames - motion with a constant velocity
- Non-inertial frames - rotational motion with constant angular velocity, acceleration in curved paths.

In the reference frame "F" which is stationary, the coordinate defined are x, y, z, and t. In another reference frame F' which moves at a velocity v which is relative to F and the observer defines coordinate in the moving reference frame as x', y', z', t'. In both the reference frames the coordinate axis are parallel and they remain mutually perpendicular. The relative motion is along the xx' axes. At  $t = t' = 0$ , the origins in both reference frames are the same  $(x,y,z) = (x',y',z') = (0,0,0)$ .

The Lorentz factor  $\gamma = 1 / (\sqrt{1 - (v^2 / c^2)})$

$$x' = \gamma(x - vt) \quad t' = \gamma(t - vx/c^2) \quad y' = y \quad z' = z$$

This is not a complete coordinate transformation since F' has to be rotated and translated so as to be co-linear with F. However, it does add the impact that relative velocity (v) has on the measurements of x and t. In most cases this impact is negligible ( $v^2 / c^2$  approaches 0). However, when v is a significant percentage of c it should be applied.

### 5.6.2. Minkowski Space Time

It is basically a combination of 3-dimensional Euclidean Space and time into a 4-dimensional manifold, where the interval of spacetime that exists between any two events is not dependent on the inertial frame of reference.

Minkowski spacetime is a 4-dimensional coordinate system in which the axes are given by (x, y, z, ct)

Where ct is time (t) times the speed of light (c)

$ds^2 = -c^2 dt^2 + dx^2 + dy^2 + dz^2$  = the differential arc length in space time where:

1. dt = change in time
2. dx = change in x direction
3. dy = change in y direction
4. dz = change in z direction

Key point - while a Lorentz transformation deals with spatial measurements, Minkowski space includes time as part of that space-time. Thus ds is an arc length through space-time as opposed to a difference in x as in the Lorentz transform.

Question, since  $c^2 dt^2$  is a negative term, does that imply that ct is an imaginary number orthogonal to

$x$ ,  $y$ , and  $z$  (cti) such that  $i^2 = -1$ ?

Yes for complex Minkowski space time. Here it is expressed as  $x^2 + y^2 + z^2 + (ict)^2 = \text{const.}$

Complex Minkowski spacetime was replaced with real Minkowski space time where time is a real coordinate rather than an imaginary one.

Where  $v$  is velocity, and  $x$ ,  $y$ , and  $z$  are Cartesian coordinates in 3-dimensional space, and  $c$  is the constant representing the universal speed limit, and  $t$  is time, the four-dimensional vector  $v = (ct, x, y, z) = (ct, \mathbf{r})$  is classified according to the sign of  $(c^2 t^2) - r^2$ . A vector is **timelike** if  $(c^2 t^2) > r^2$ , **spacelike** if  $(c^2 t^2) < r^2$ , and **null** or **lightlike** if  $(c^2 t^2) = r^2$ . This can be expressed in terms of the sign of  $\eta(v, v)$  as well, which depends on the signature. The classification of any vector will be the same in all frames of reference that are related by a Lorentz transformation (but not by a general Poincaré transformation because the origin may then be displaced) because of the invariance of the interval.

# Annex A: Bibliography

Hughes, P. :Spacecraft Attitude Dynamics, Dover Publications Inc. 2024

ISO/TC 211: ISO 19107:2019 Geographic information - Spatial schema

ISO/TC 211: ISO 19107:2003 Geographic information - Spatial schema

ISO/TC 211: ISO 19109:2005 Geographic information - Rules for application schema

ISO/TC 211: ISO 19109:2005 Geographic information - Rules for application schema

ISO/TC 211: ISO 19141:2008 Geographic information - Schema for moving features

OGC: OGC 20-010, OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard (2021)

OGC: OGC 19-011r4, OGC IndoorGML 1.1 (2020)

OGC: OGC 19-045r3, OGC MOving Features Encoding Extension - JSON (2020)



## Annex B: Revision History

Date	Release	Editor	Primary clauses modified	Description
2016-04-28	0.1	G. Editor	all	initial version