Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: 2022-03-15

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

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

Table of Contents

1. Scope 5
2. References 6
3. Terms and Definitions 8
    3.1. **term name** 8
4. Conventions 9
    4.1. Identifiers 9
5. Dynamic Features 10
    5.1. Geometry in 3 Dimensions 10
        5.1.1. Volumetrics 11
        5.1.2. Boundaries 12
        5.1.3. Closure Surfaces 14
        5.1.4. B-Rep 15
        5.1.5. Point Clouds 15
        5.1.6. SPICE 15
    5.2. Moving Features 15
        5.2.1. Basic Concepts 16
        5.2.2. MF_OneParameterGeometry 17
        5.2.3. MF_TemporalGeometry 18
        5.2.4. Temporal Properties 18
        5.2.5. MF_TemporalTrajectory 19
        5.2.6. MF_PrismGeometry 20
        5.2.7. Rigid vs. Plastic 23
    5.3. Temporal Properties 25
        5.3.1. TemporalProperties Object 25
    5.4. Articulated Geometries 26
        5.4.1. GeoPOSE 26
        5.4.2. MISB Staging System 26
        5.4.3. Discussion 28
        5.4.4. Lorentz Transformation 29
        5.4.5. Minkowski Space Time 29
    5.5. Complex Articulated Systems 30
    5.6. Mass Properties 30
    5.7. The Problem of Time 30
    5.8. Temporal Reference Systems 32
    5.9. Dynamic Features at Relativistic Velocities 32
Annex A: Revision History 34
Annex B: Bibliography 35

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

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

NOTE

Insert References here. If there are no references, state “There are no normative references”.

References are to follow the Springer LNCS style, with the exception that optional information may be appended to references: DOIs are added after the date and web resource references may include an access date at the end of the reference in parentheses. See examples from Springer and OGC below.

Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)

May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)

Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)

Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–184. IEEE Press, New York (2001)

Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002)

National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov

ISO / TC 211: ISO 19115-1:2014 Geographic information — Metadata — Part 1: Fundamentals (2014)

ISO / TC 211: ISO 19157:2013 Geographic information — Data quality (2013)

ISO / TC 211: ISO 19139:2007 Geographic information — Metadata — XML schema implementation (2007)

ISO / TC 211: ISO 19115-3: Geographic information — Metadata — Part 3: XML schemas (2016)

OGC: OGC 15-097 OGC Geospatial User Feedback Standard. Conceptual Model (2016)

OGC: OGC 12-019, OGC City Geography Markup Language (CityGML) Encoding Standard (2012)

OGC: OGC 14-005r3, OGC IndoorGML (2014)

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. term name

text of the definition

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

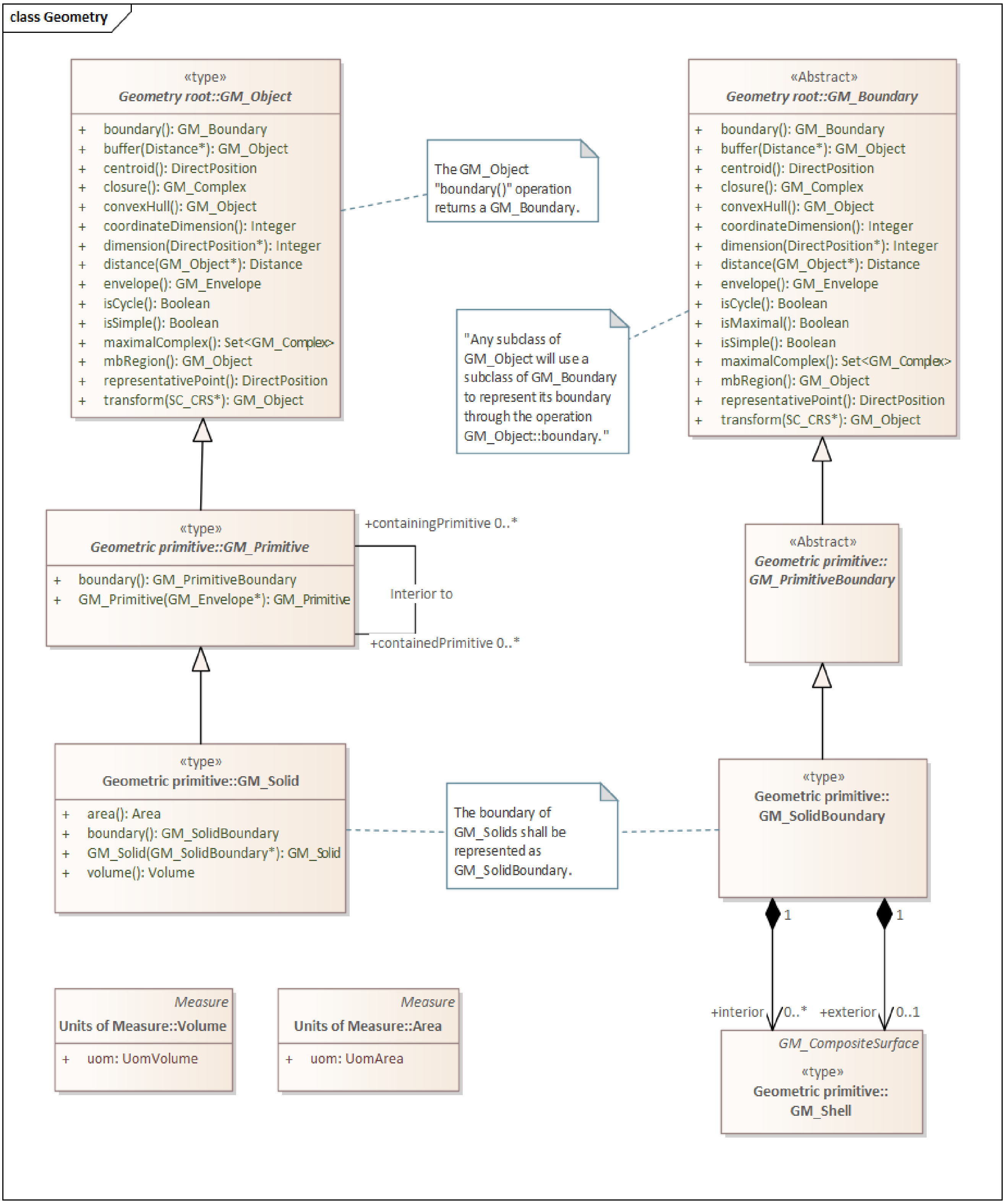
5.1. Geometry in 3 Dimensions

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

First we need to define 3D. Our proposed definition is as follows:

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

This definition is consistent with ISO 19107. This ISO standard defines both the GM_Object and GM_Boundary classes as root level geometry classes. The association between object and boundary is achieved through the "boundary()" operation on the GM_Object class.



GM_Object and GM_Boundary are dimensionless. Further refinement is needed to fully define a 3D geometry.

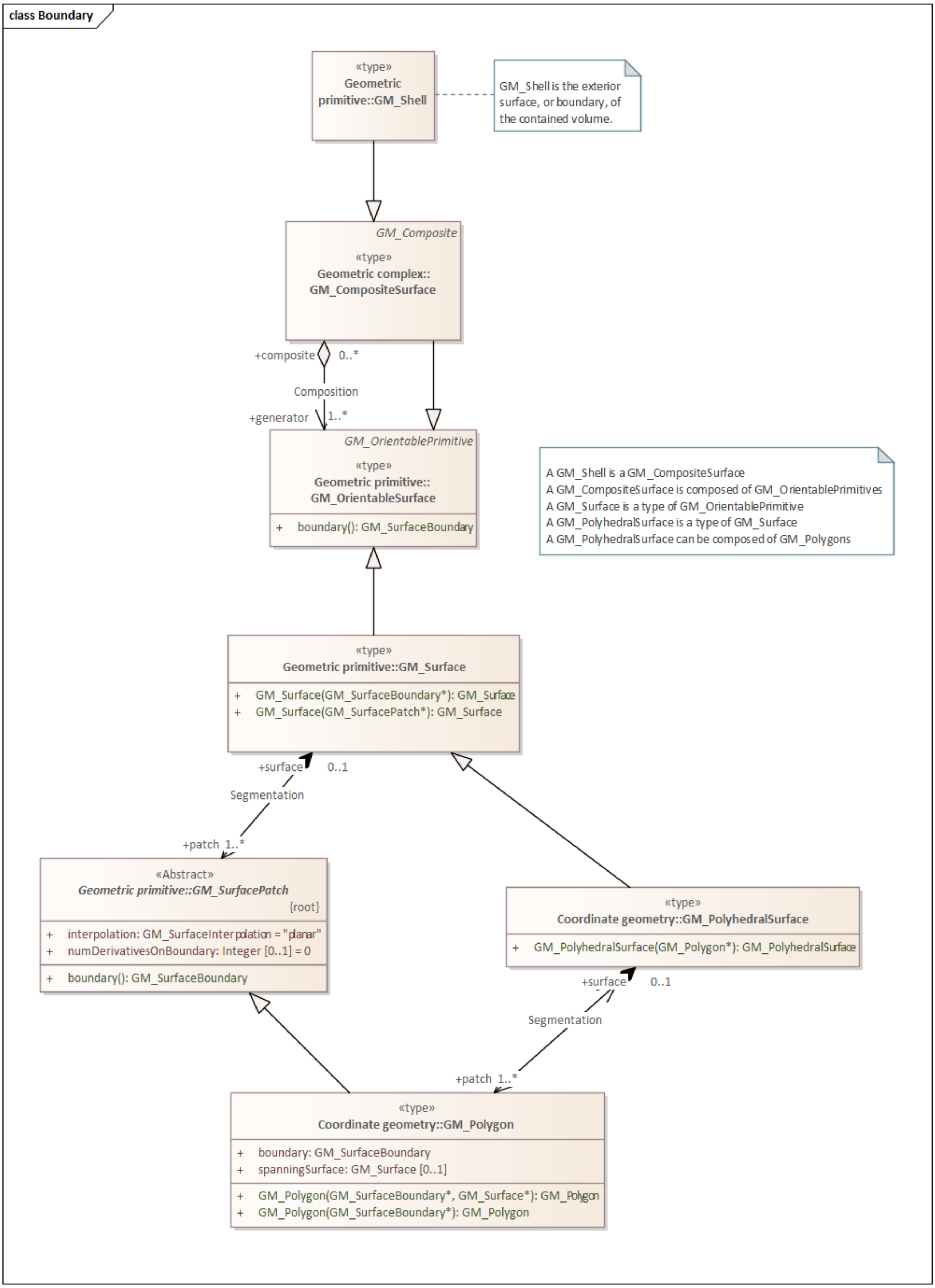
5.1.1. Volumetrics

Starting with 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.

5.1.2. 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:

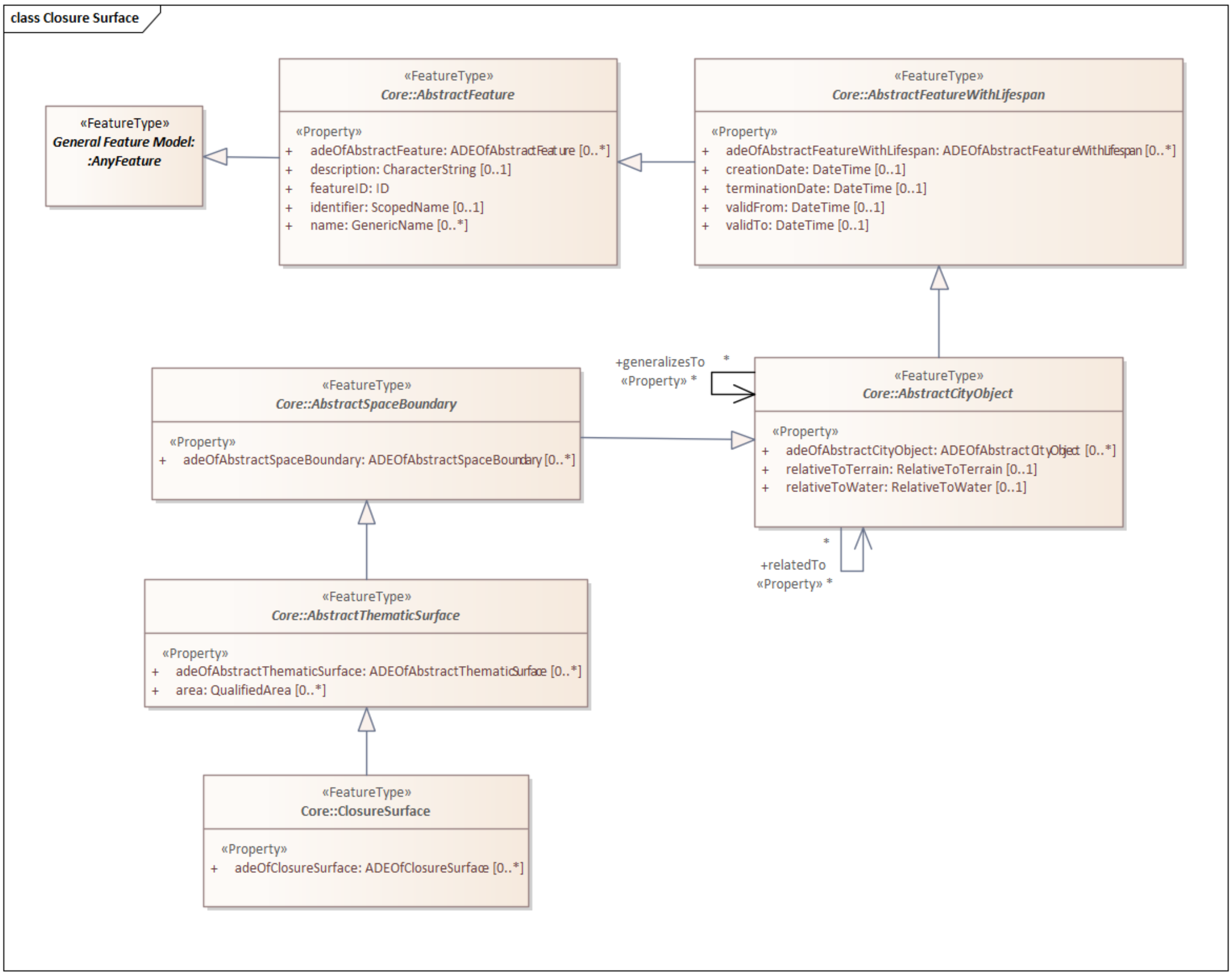
- A GM_Shell is a GM_CompositeSurface
- A GM_CompositeSurface is composed of GM_OrientablePrimitives
- A GM_Surface is a type of GM_OrientablePrimitive
- A GM_PolyhedralSurface is a type of GM_Surface
- A GM_PolyhedralSurface can be composed of GM_Polygons

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)

5.1.3. 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 desireable 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 ancestory. We may want to generalize this concept for use outside of CityGML. However, the capabilties provided by the ancestor classes do provide value and may be worth incorporating into a general 3D model.

5.1.4. 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

5.1.5. 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."

5.1.6. SPICE

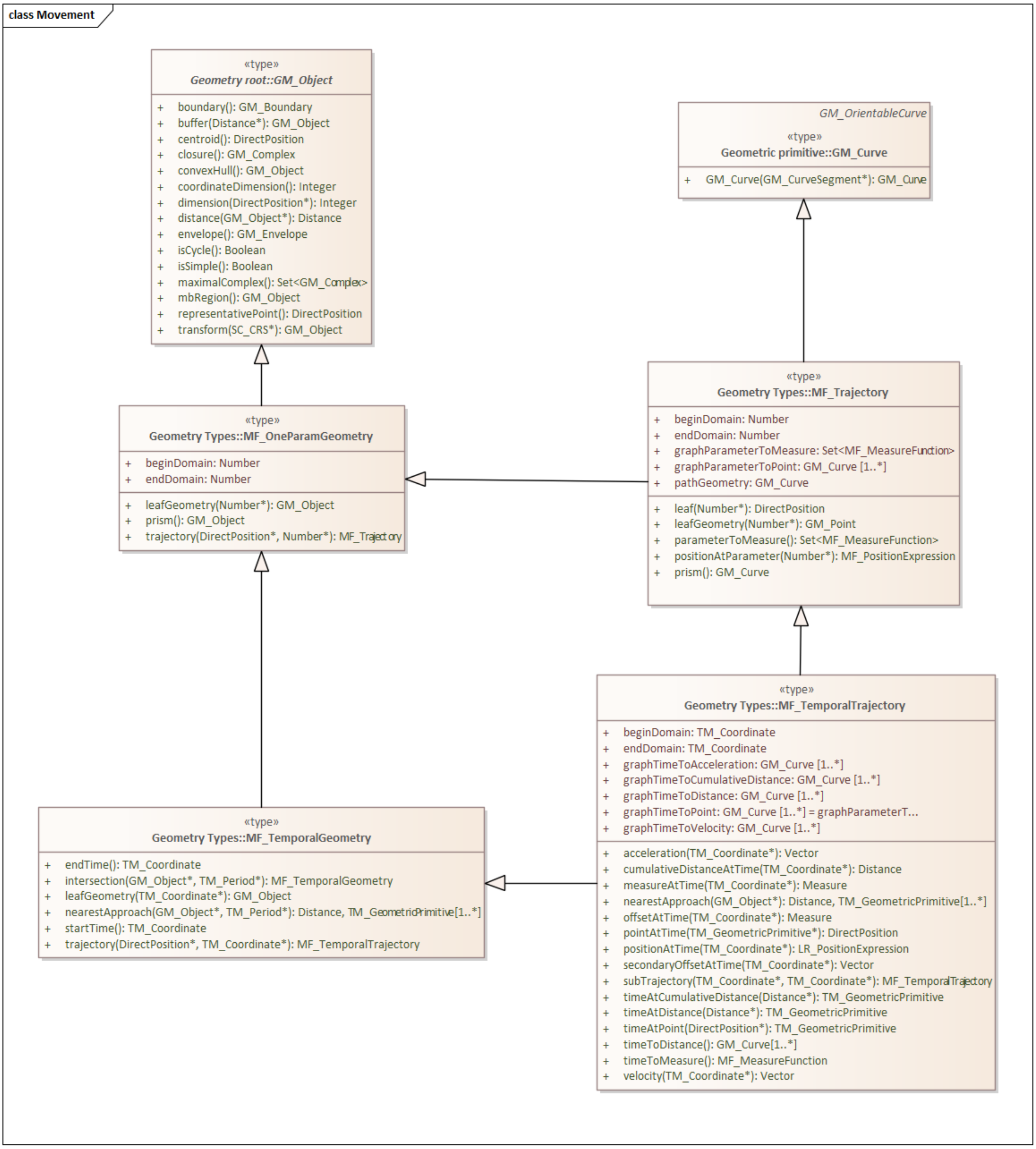
https://naif.jpl.nasa.gov/naif/toolkit.html

This section will contain a discussion of SPICE and how well it aligns with the ISO/OGC concepts.

5.2. Moving Features

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

ISO 19141 is the standard for Moving Features. It introduces the concept of Feature Properties whose values are a function of an input parameter.



5.2.1. Basic Concepts

Volume and Boundary surface

ISO 19107 provides us with the GM_Object and GM_Boundary classes. GM_Object represents the space that is occupied by a geometry. GM_Boundary represents a surface which forms the boundary between inside and outside of the associated GM_Object. GM_Object and GM_Boundary are both specialized according to the number of dimensions they support.

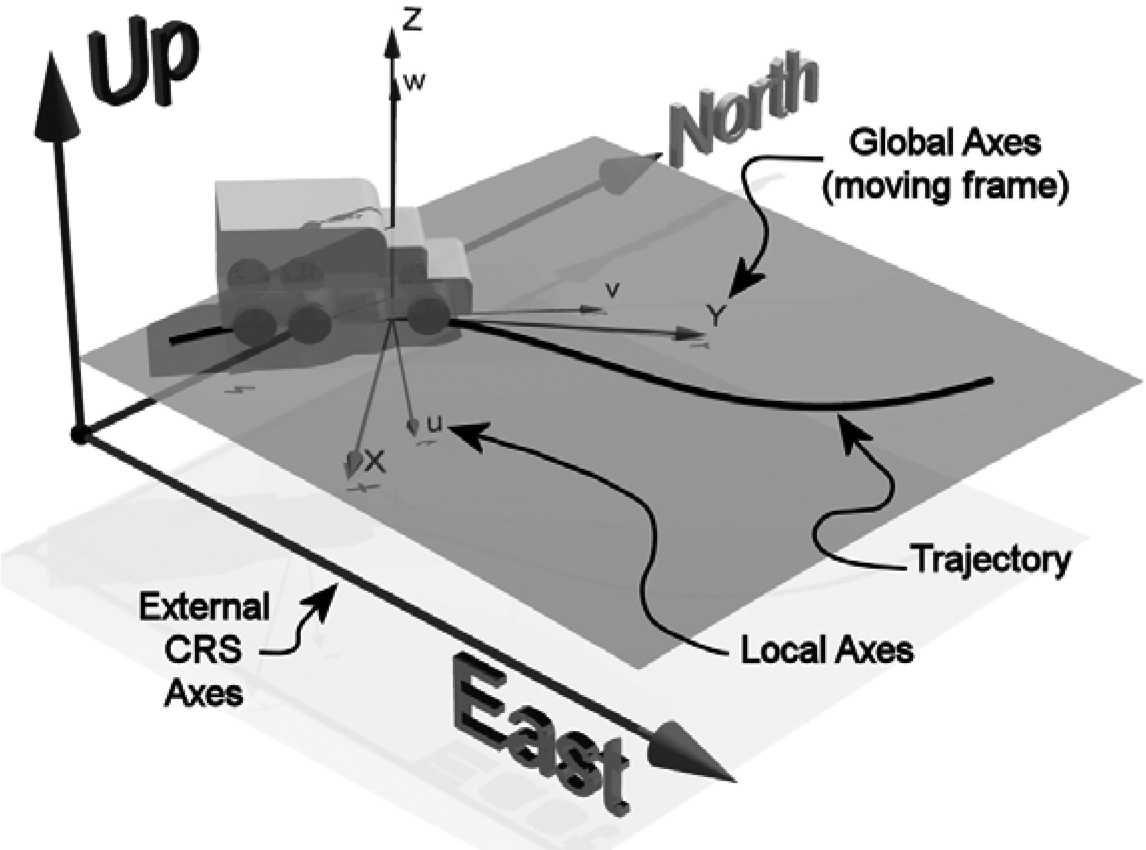
Location and shape

The location of a Feature is a single GM_Point which represents its current position in the universe.

GM_Point is a geometry of zero dimensions, but is specified with values of all of the applicable coordinate axis.

The Shape of a Feature is the bounding geometry. Shape tells you nothing about where a Feature is but does specify the volume of the space it occupies.

Global and Local coordinate systems



Moving Features deal with two coordinate system. The Global (external) coordinate system is defined for the area external to the Feature. It is mostly used to identify the loctation and orientation of the Feature.

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

5.2.2. 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 Î [a, b] such that f(t) is a geometry and for each point P Î f(a) there is a one parameter set of points (called the trajectory of P) P(t) : [a, b] ® P(t) such that P(t) Î 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.

5.2.3. MF_TemporalGeometry

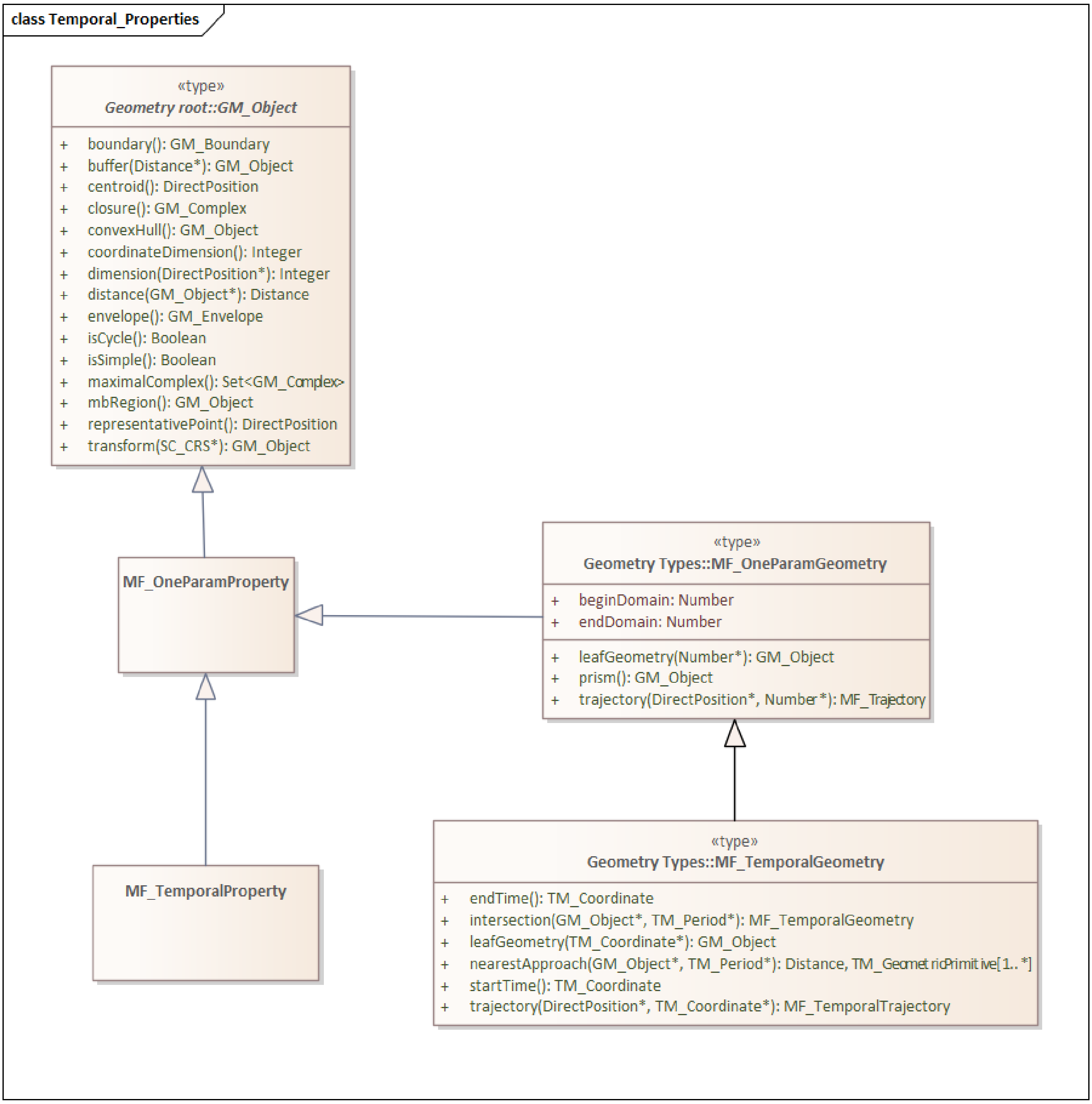
An MF_TemporalGeometry is a MF_OneParameterGeometry where the parameter is Time expressed as a TM_Coordinate. TM_Coordinate is specified in ISO 19108; it expresses time as a multiple of a single unit of measure such as year, day, or second. The "leafgeometry()" operation of an instance of MF_TemporalGeometry would take a TM_Coordinat in as input and return a GM_Object instance representing the geometry of the Feature at the specified point in time.

5.2.4. 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."

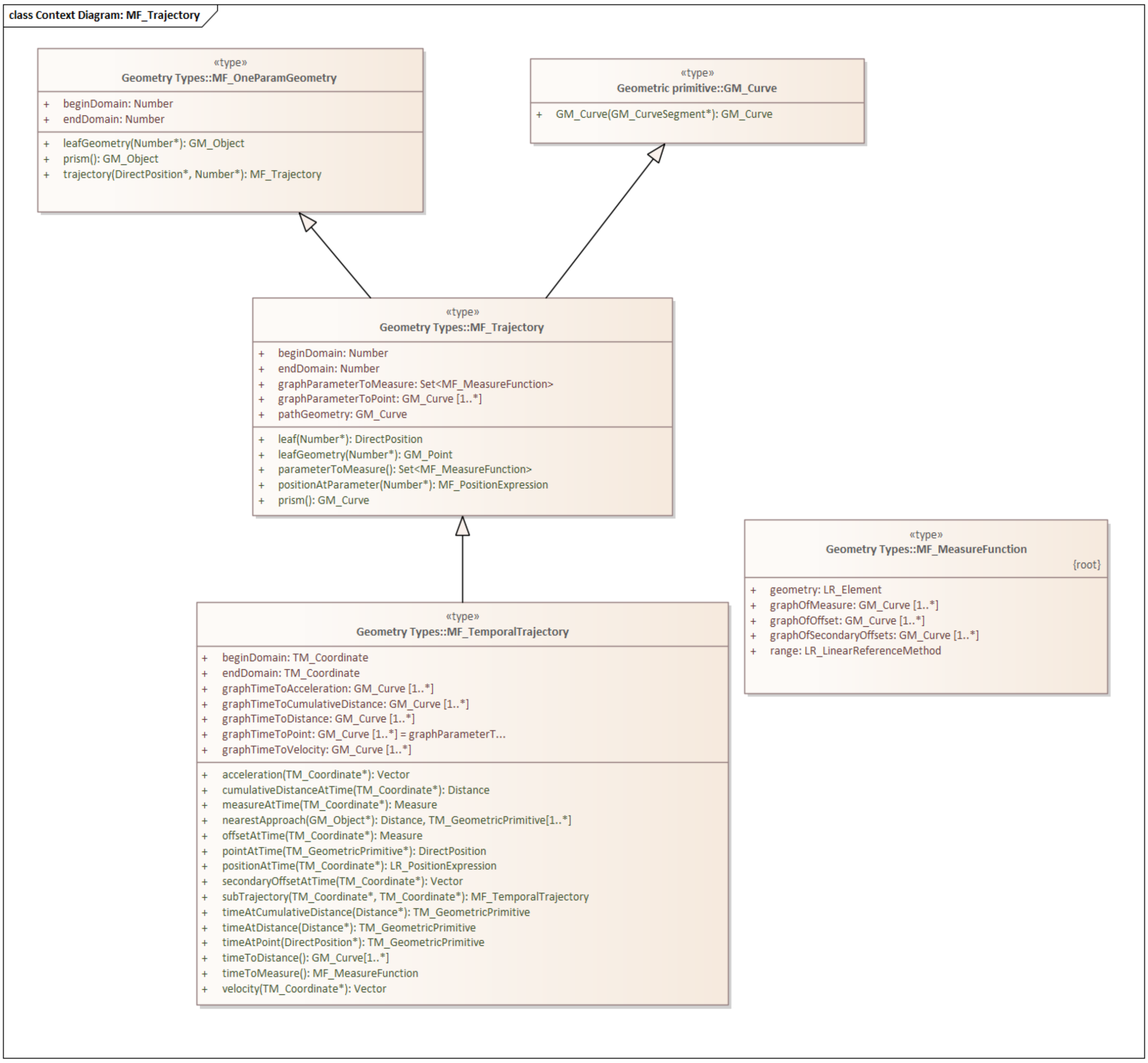
Logically TemporalProperties should be a subclass of OneParamProperties. Since Geometry is a property, then MF_TemporalGeometry should be a subclass of TemporalProperties. Which gives us the following UML.



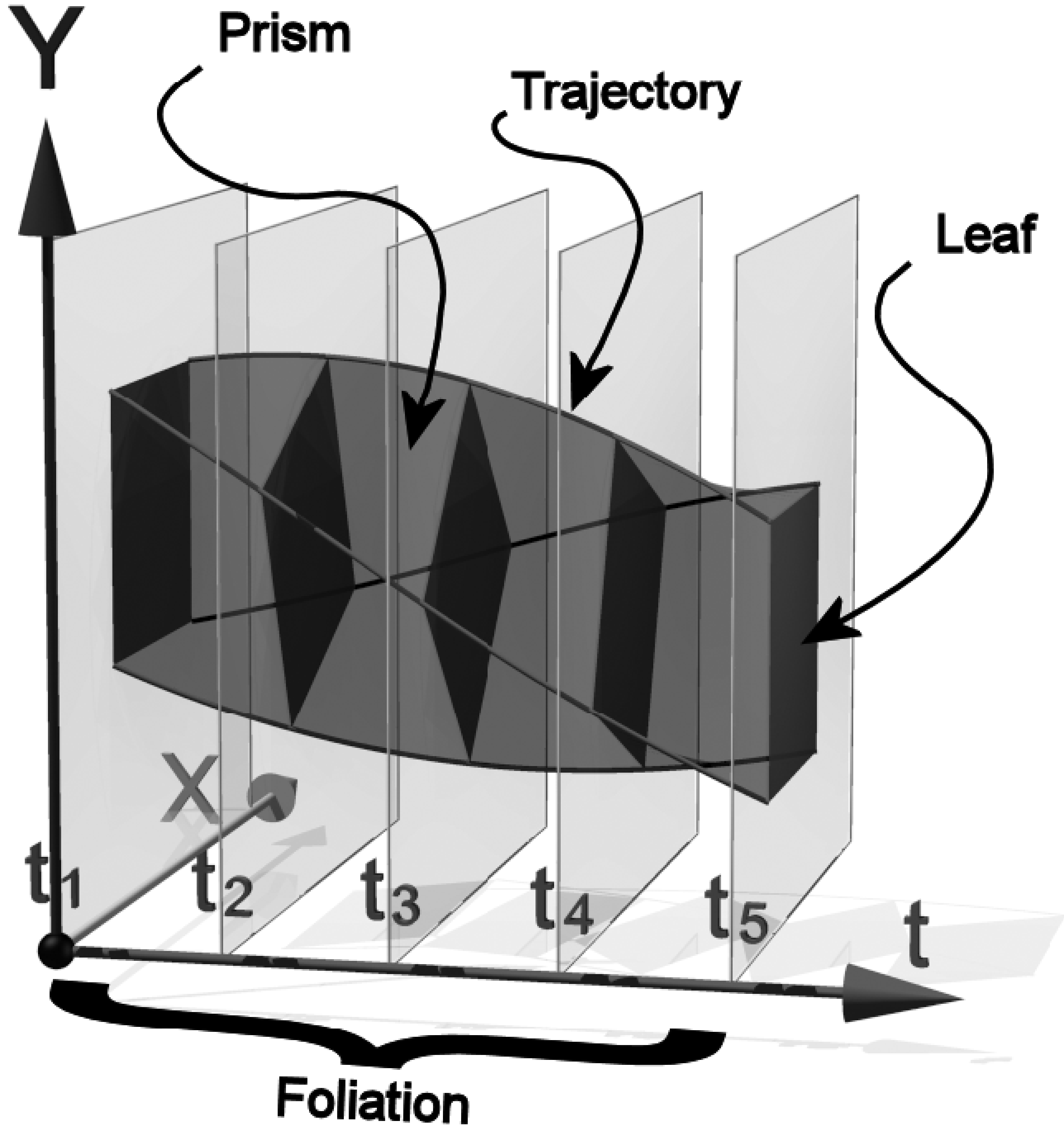
Temporal properties are particularly useful for capturing state change. For example, the fuel load of an aircraft will change over time. The leafproperty() operation on a temporal fuel_load object would return the amount of fuel onboard at the specified time.

5.2.5. MF_TemporalTrajectory

A trajectory is a GM_Curve which represents the positions a Moving Feature has or is expected to occupy. A Temporal Trajectory supports operations which will report the properties of the Moving Feature at a specific time. The geometry property is restricted to GM_Point.



5.2.6. MF_PrismGeometry



A leaf represents the geometry of a Feature as a function of time

Each point in the bounding surface is traced across leafs using a temporal trajectory

The space occupied by the Feature over time, the sum of all of the leaf geometries, is a Prism.

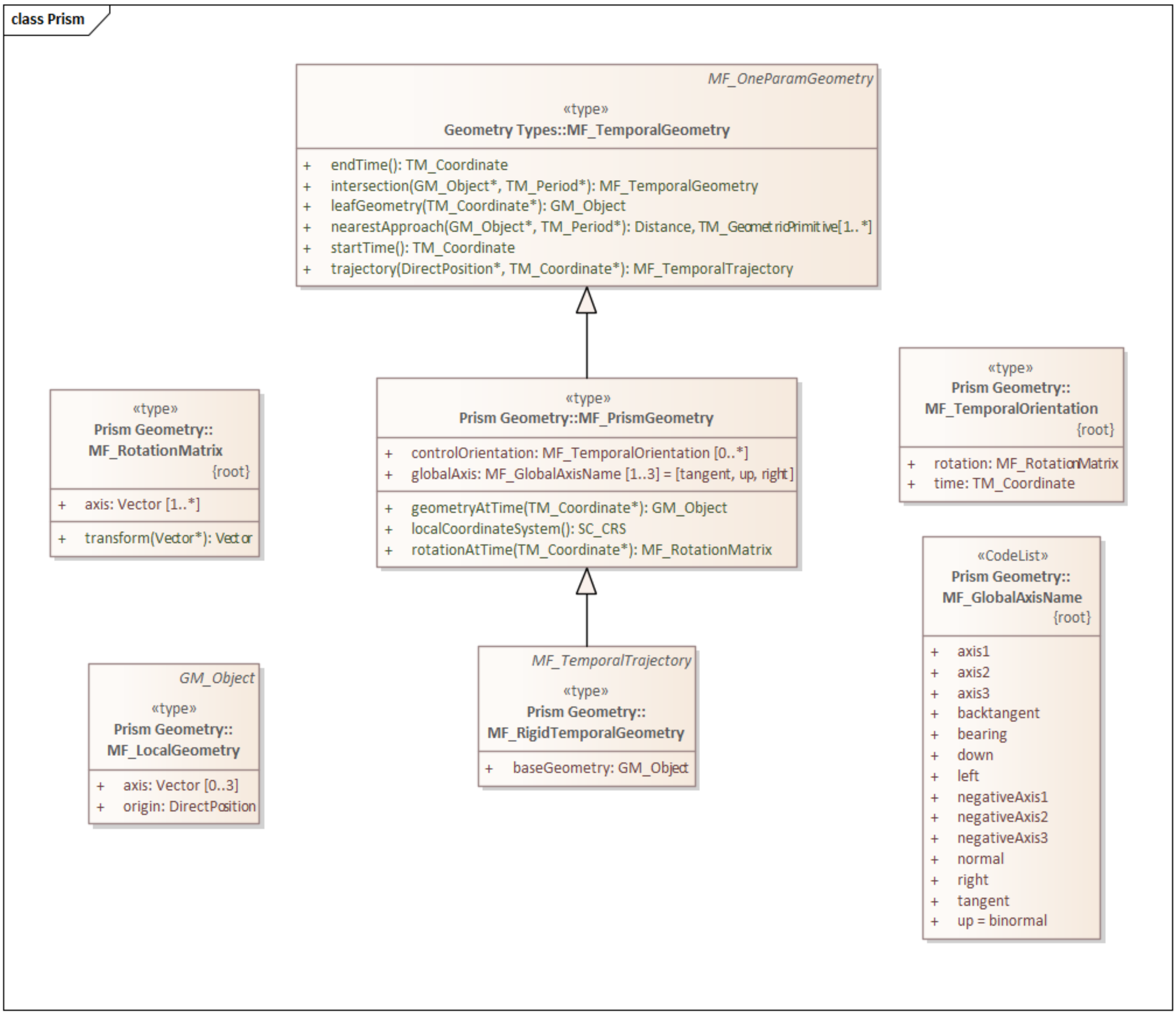
A collection of leafs is a foliation.

1. A leaf is the same in both the prism and trajectory model. The difference is in the numbmer of dimensions allowed
2. A prism can be viewed as a volume enclosed by a surface defined by a collection of trajectories

This gives up two alternatives. A leaf can store the full geometry as it exists at the specified time. Or, a leaf can generate the geometry from the collection of trajectories.

In the first case, the full geometry must be represented in each leaf whether or not it has changed. However, it is most easily represented in local coordinate space so that only actual changes in the shape have to be captured.

In the second case intermediate geometries do not have to be captured. The values at a specific time can be calculated from the trajectory curve. However, for a moving feature it becomes necessary to update all fo the trajectories whenever the feature changes location.



GeometryAtTime(): The operation "geometryAtTime" shall accept a time in the domain of the prism geometry and return the geometry of the moving feature, as it is at a given time in the global coordinate reference system. This shape might be a realistic rendition of the object, or it may be an iconized rendition of the type of object, as needed by the application. For example, in a simulation a truck might be represented as an icon as opposed to a photorealistic rendition. This allows the application to use the local geometry to convey information such as certainty of identification or feature status through the use of appropriate icons and other portrayal parameters.

RotationAtTime(): The operation "rotationAtTime" shall accept a time in the domain of the prism geometry and return 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 "map" coordinate reference system. Because scale may change in complex ways as a feature moves with respect to a projected coordinate reference system, the rotation matrix may also contain scale factors.

LocalCoordinateSystem(): The operation "localCoordinateSystem" shall return a SC_CRS for the design coordinate reference system in which the moving feature's shape is defined.

BaseGeometry Attribute: The attribute "baseGeometry" is the geometry of the moving object in a local rectangular coordinate reference system based on the axis of the object. The object should have a natural up, front and right (cross product of up and front). If the representation to be used is not "centred" on its origin, the application should use an MF_LocalGeometry (7.3) subclass to create a local origin.

5.2.7. Rigid vs. Plastic

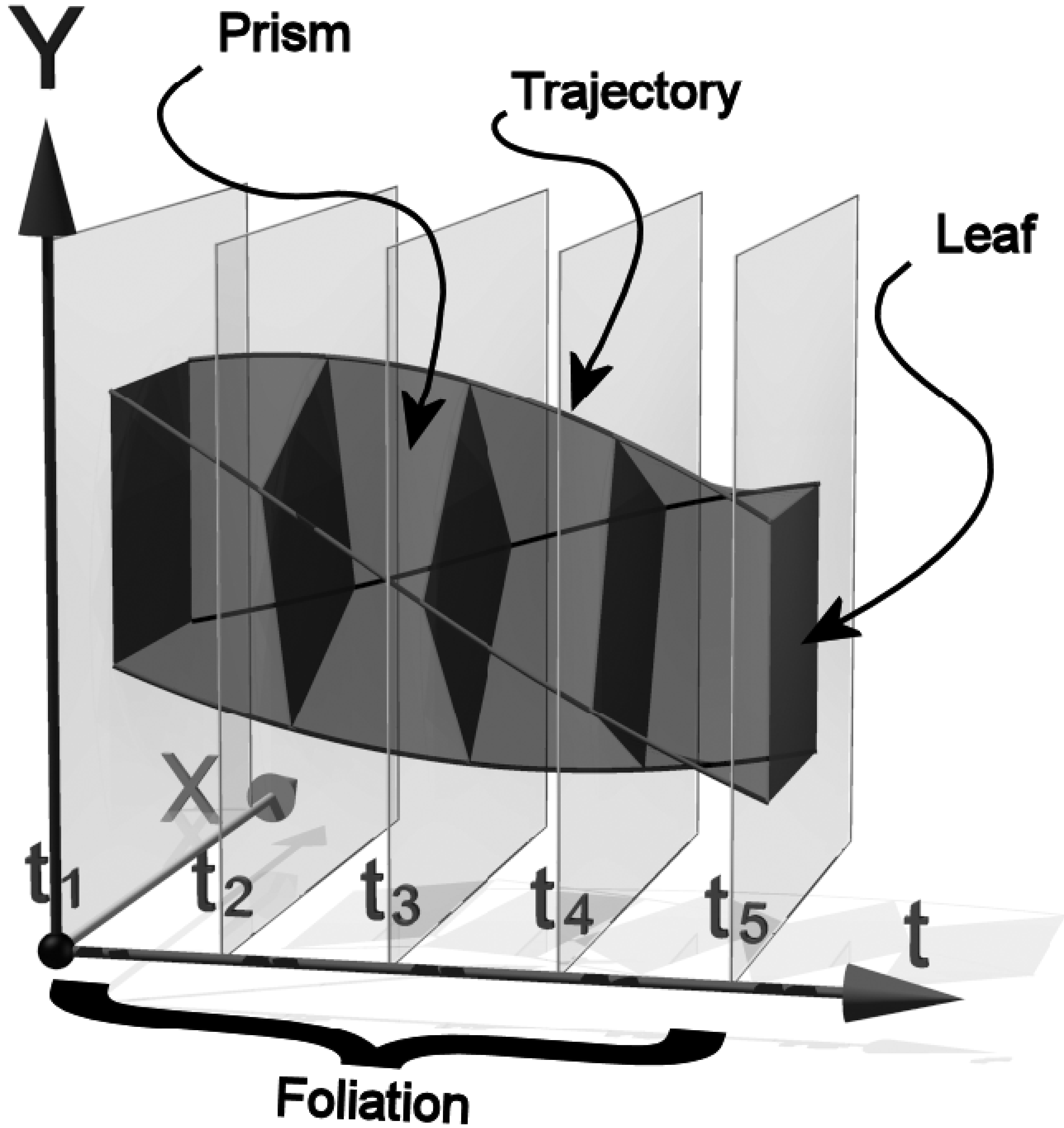
ISO 19141:2008 states that it "defines a method to describe the geometry of a feature that moves as a rigid body."

That may not be a limitation in fact. If our object iS defined by a bounding surface, and that surface is made up of a number of shapes descended from MF_TemporalGeometry, then the surface geometry of our object can change over time. Hence it is not rigid. ISO 19141 can also define deforming (plastic) bodies.

ISO 19141 Section 5.1

A moving feature can be modelled as a combination of movements. The overall motion can be expressed as the temporal path or trajectory of some reference point on the object (the "origin"), such as its center of gravity. Once the origin's trajectory has been established, the position along the trajectory can be described using a linear reference system (as defined in ISO 19133). The "parameterization by length" for curves (as defined in ISO 19107) can be used as a simple linear reference if no other is available. The relationship between time (t) and measure value (m) can be represented as the graph of the t→ m function in a plane with coordinates (t, m). This separation of the geometry of the path and the actual "time to position" function allows the moving feature to be tracked along existing geometry.

Figure 1 illustrates how the concepts of foliation, prism, trajectory, and leaf relate to one another. In this illustration, a 2D rectangle moves and rotates. Each representation of the rectangle at a given time is a leaf. The path traced by each corner point of the rectangle (and by each of its other points) is a trajectory. The set of points contained in all of the leaves, and in all of the trajectories, forms a prism. The set of leaves also forms a foliation.



These two object representations, of the path and the position along that path, give the general position of the moving feature. The other variable in describing the position of the feature is the rotation about the chosen reference point. To describe this, a local engineering coordinate system is established using the object reference point as its origin. The geometry of the feature is described in the engineering coordinate system and the real-world orientation of the feature is given by mapping of the local coordinate axes to the global coordinate system (the CRS of the trajectory of the reference point). This can be given as a matrix that maps the unit vectors of the local coordinate system to vectors in the global CRS.

If the global CRS and local CRS have the same dimension, then each point within the local CRS can be traced in time through the global CRS by combinations of these various mappings. The map would trace from time (t) to the measure (m) to a position on the reference point's path using the LRS. Then using the rotation matrix, the calculated offset from this point gives a direct position in the global CRS.

This means that the ‘prism’ of the moving feature (defined as all the points which part of the feature passes through) can be viewed (and calculated to whatever degree of accuracy needed) as a bundle of trajectories of points on the local engineering representation of the feature’s geometry. If viewed in a 4 dimensional spatiotemporal coordinate system, the points on the feature at different times are different points. Then the preimage of the prism (points on the trajectories augmented by a time coordinate) is a foliation, meaning that there is a complete and separate representation of the geometry of the feature for each specific time (called a “leaf”). These names come from a 3D metaphor of a book, where each page or leaf is a slice of time in the “folio”.

This might form the basis for an extension of this standard to non-rigid, mutable objects. Each leaf in the 4D foliation is a separate representation of the object, and by creating methods to describe the change through time of the shape and form of the feature, the existing machinery in this International Standard can be used to place those representations in positions with respect to the global coordinate system.

5.3. Temporal Properties

Proposition: Features can change in state as well at location

ISO 19141 introduced the concept of a One Parameter Geometry.

MF_OneParamGeometry:

A one parameter set of geometries is a function f from an interval t Î [a, b] such that f(t) is a geometry and for each point P Î f(a) there is a one parameter set of points (called the trajectory of P) P(t) : [a, b] ® P(t) such that P(t) Î f(t). A leaf of a one parameter set of geometries is the geometry f(t) at a particular value of the parameter. The set of geometries forms a prism that is the set of points in the union of the geometries (or the union of the trajectories).

The type "MF_OneParamGeometry" acts as the root classifier for all geometric object classified as one parameter sets of geometric objects. As a one-paramter set, they have "leaf" projections (for each parameter value. As geometric objects they act as the infinite union of all their "leaf" (cross sections for each parameter value).

According to the Feature Model, geometry is a property of a Feature. Since we have defined MF_OneParamGeometry, it stands to reason that there can be a MF_OneParamProperty. Likewise, there can be a subclass MF_TemporalProperty.

Therefore, MF_TemporalProperty can represent any Feature Property which changes with time. The JSON encoding standard for Moving Features provides us whith this capability.

5.3.1. TemporalProperties Object

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.

ParametricValues Object

A ParametricValues object is a JSON object that represents a collection of parametric values of dynamic non-spatial attributes that are ascertained at the same times. A parametric value may be a time-varying measure, a sequence of texts, or a sequence of images. Even though the parametric value may depend on the spatiotemporal location, MF-JSON Prism only considers the temporal dependencies of their changes of value.

5.4. 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.

5.4.1. 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

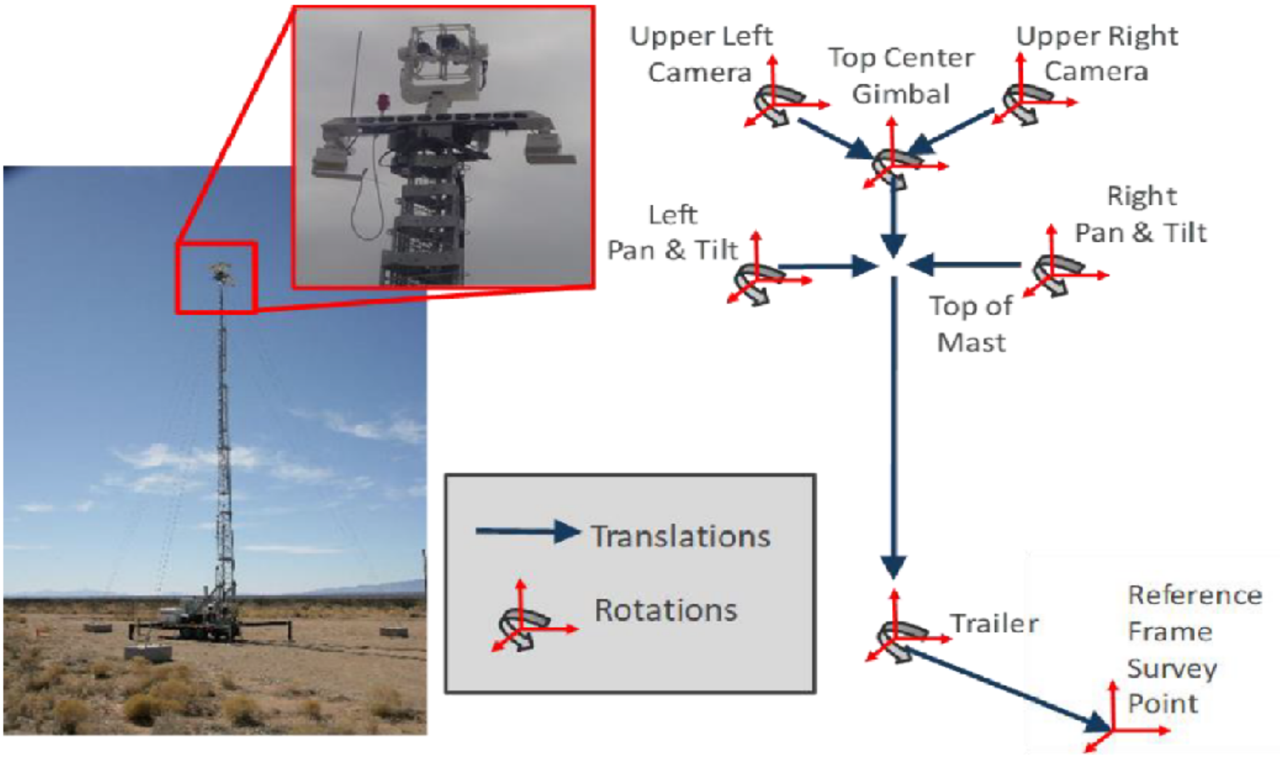
The sequence logical model defines a method for packaging of GeoPose data, where multipleGeoPoses in a sequence share the same Outer Frame and there is a time-dependent changing Inner Frame.

5.4.2. 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 instateneous 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 kinemantics 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

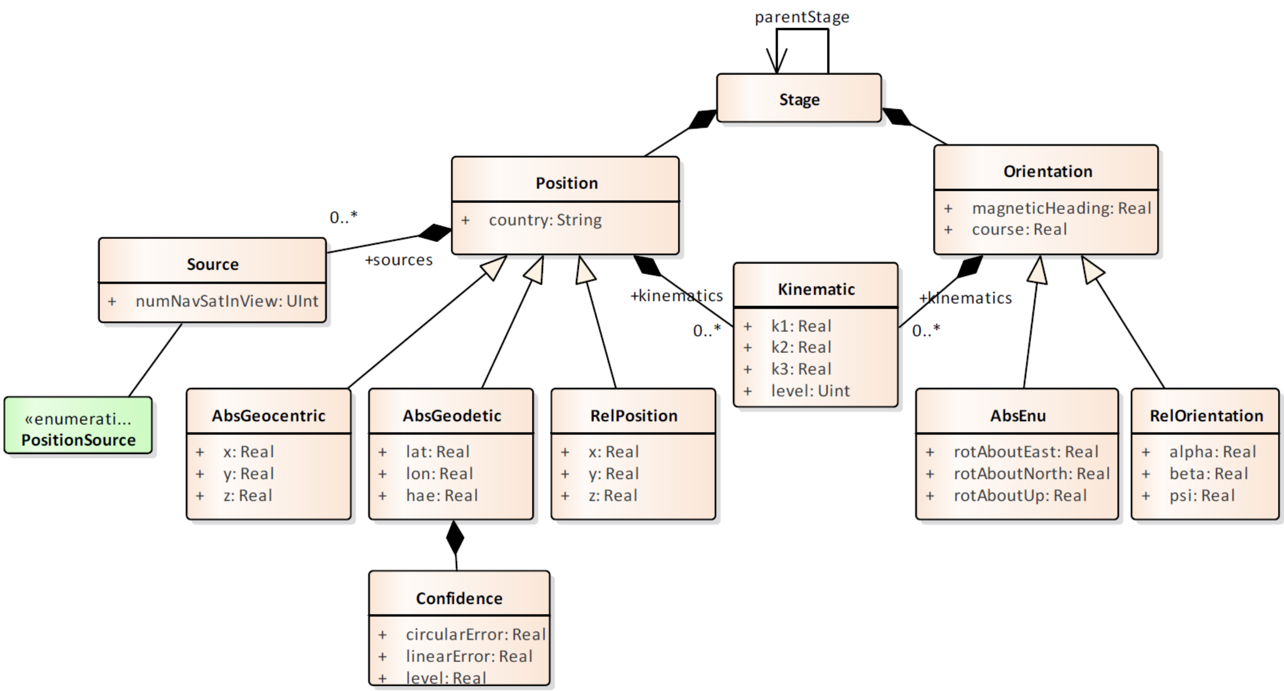
Stage	Component	Parent	Values
1	Reference Frame Survey Point	0	Absolute Position/Orientation
2	Trailer	1	Relative Position / Orientation
3	Left Camera	2	Relative Position / Orientation
4	Right Camera	2	Relative Position / Orientation
5	Top Center Gimbal	2	Relative Position / Orientation
6	Left Upper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

5.4.3. Discussion

Quaternions vs. Euler rotations

Temporal representation

Standarize on a right-handed Cartisian coorinate 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

5.4.4. 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 reverence frame as x', y', z', t'. In both the reference frames the coordinate axis are parallel and they remain mutually perpedicular. 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 g = 1 / (sqrt(1 - (v^2 / c^2)))

x' = g(x-vt) t' = g(t-(vx/c^2)) y' = y 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 neglegible (v^2 / c^2 approaches 0). However, when v is a significant percentage of c it should be applied.

5.4.5. 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^{2dt}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^{2dt}2 is a negative term, does that inply that ct is an imaginary number orthagonal 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 = 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, **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 η(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.

5.5. Complex Articulated Systems

This may be a redundant section.

5.6. Mass Properties

This may be a redundant section.

5.7. The Problem of Time

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 existance. Yet the concepts of time used in the geospatial community are almost exclusivly 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 TRS. A temporal equivalent to GeoPOSE.

Introduce the Timer concept from MISB.

GM_Object

Operation	Parameters	Returns
boundary		GM_Boundary

buffer	Distance	GM_Object
centroid		DirectPosition
closure		GM_Complex
convexHull		GM_Object
coordinateDimension		Integer
dimension	DirectPosition	Integer
distance	GM_Object	Distance
envelope		GM_Envelope
isCycle		Boolean
isSimple		Boolean
maximalComplex		set<GM_Complex>
mbRegion		GM_Object
represenativePoint		DirectPosition
transform	SC_CRS	GM_Object

MF_OneParamGeometry

Operation	Parameters	Returns
leafGeometry	Number	GM_Object
prism		GM_Object
trajectory	DirectPosition, Number	MF_Trajectory

MF_TemporalGeometry

Operation	Parameters	Returns
endTime		TM_Coordinate
intersection	GM_Object, TM_Period	TM_TemporalGeometry
leafGeometry	TM_Coordinate	GM_Object
nearestApproach	GM_Object, TM_Period	Distance, TM_GeometricPrimitive
startTime		TM_Coordinate
trajectory	DirectPosition, TM_Coordinate	MF_TemporalTrajectory

MF_Trajectory

Operation	Parameters	Returns
leaf	Number	DirectPosition
leafGeometry	Number	GM_Point

parameterToMeasure		set<MF_MeasureFunction>
positionAtParameter	Number	MF_PositionExpression
prism		GM_Curve

MF_TemporalTrajectory

Operation	Parameters	Returns
acceleration	TM_Coordinate	Vector
cumulativeDistanceAtTime	TM_Coordinate	Distance
measureAtTime	TM_Coordinate	Measure
nearestApproach	GM_Object	Distance, TM_GeometricPrimitive
offsetAtTime	TM_Coordinate	Measure
pointAtTime	TM_GeometricPrimitive	DirectPosition
positionAtTime	TM_Coordinate	LR_PositionExpression
secondaryOffsetAtTime	TM_Coordinate	Vector
subTrajectory	TM_Coordinate, TM_Coordinate	MF_TemporalTrajectory
timeAtCumulativeDistance	Distance	TM_GeometricPrimitive
timeAtDistance	Distance	TM_GeometricPrimitive
timeAtPoint	DirectPosition	TM_GeometricPrimitive
timeToDistance		GM_Curve
timeToMeasure		MF_MeasureFunction
velocity	TM_Coordinate	Vector

MF_PrismGeometry

Operation	Parameters	Returns
geometryAtTime	TM_Coordinate	GM_Object
localCoordinateSystem		SC_CRS
rotationAtTime	TM_Coordinate	MF_RotationMatrix

5.8. Temporal Reference Systems

May be redundant

5.9. Dynamic Features at Relativistic Velocities

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

More to come.

Annex A: Revision History

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

Annex B: Bibliography

NOTE

Example Bibliography (Delete this note).

The TC has approved Springer LNCS as the official document citation type.

Springer LNCS is widely used in technical and computer science journals and other publications

- For citations in the text please use square brackets and consecutive numbers: [1], [2], [3]

– Actual References:

[n] Journal: Author Surname, A.: Title. Publication Title. Volume number, Issue number, Pages Used (Year Published)

[n] Web: Author Surname, A.: Title, http://Website-Url

[1] OGC: OGC Testbed 12 Annex B: Architecture. (2015).