OGC® DOCUMENT: 22-005

External identifier of this OGC® document: http://www.opengis.net/doc/DP/DynamicFeatures/{1.0}

Open Geospatial Consortium

DISCUSSION PAPER FOR OGC DYNAMIC FEATURES

DISCUSSION PAPER

DRAFT

Submission Date: 2022-11-28
Approval Date: 2019-02-11
Publication Date: 2019-12-11
Editor: Charles Heazel

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.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Copyright notice

Note

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.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

I KEYWORDS

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

ogcdoc, OGC document, OGC, IndoorGML, Indoor space, Outdoor space, Seamless navigation, CityGML

II PREFACE

i. Abstract

Moving Feature standards and technologies have made considerable progress. As they mature, implementations will move beyond our traditional Earth-centric framework. Extraterrestrial, deep space, and even virtual deployment environments must be considered.

It is time to re-evaluate the foundations of OGC standards to determine if they are sufficient to support these emerging deployment environments. 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

III SECURITY CONSIDERATIONS

No security considerations have been made for this document.

IV SUBMITTING ORGANIZATIONS

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

- Heazeltech LLC

1 SCOPE

1 SCOPE

Moving Feature standards and technologies have made considerable progress in recent years. As they mature, implementations will move beyond our traditional Earth-centric framework. Extraterrestrial, deep space, and even virtual deployment environments must be considered.

This paper focuses on Features which have a full three-dimensional geometry and are capable of movement, potentially at very high velocities. In addition, these Features may be independent of the Earth. WGS-84 and the Gregorian Calender have little relevance to a Feature flying through the Oort Cloud. Even less so for Features in artificial virtual worlds. Features with this expanded scope challenge the assumptions underlying both the OGC and ISO TC211 standards.

This paper provides an analysis of the existing OGC Standards baseline, seeking to assess how well it supports these new deployments and identifies any future work required.

2

NORMATIVE REFERENCES

2 NORMATIVE REFERENCES

There are no normative references in this document.

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

3

TERMS AND DEFINITIONS

3 TERMS AND DEFINITIONS

No terms and definitions are listed in this document.

As a Discussion Paper, this document contains no normative definitions. The terms and definitions used in this discussion paper are documented in the Glossary (Annex C).

4

CONVENTIONS

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.

4.2. UML Notation

The logical structure of the elements used in this Discussion Paper is presented using the Unified Modeling Language (UML) static structure diagram (see Booch et al. 1997). The UML notations used in this paper are described in the diagram in Figure 1.

Most of these UML diagrams have been extracted from the ISO/TC211 Harmonized Model maintained by the Harmonized Model Maintenance Group on their GitHub Repository. Some modifications may have been made to better present issues relevant to this discussion. None of these modifications change the underlying UML model.

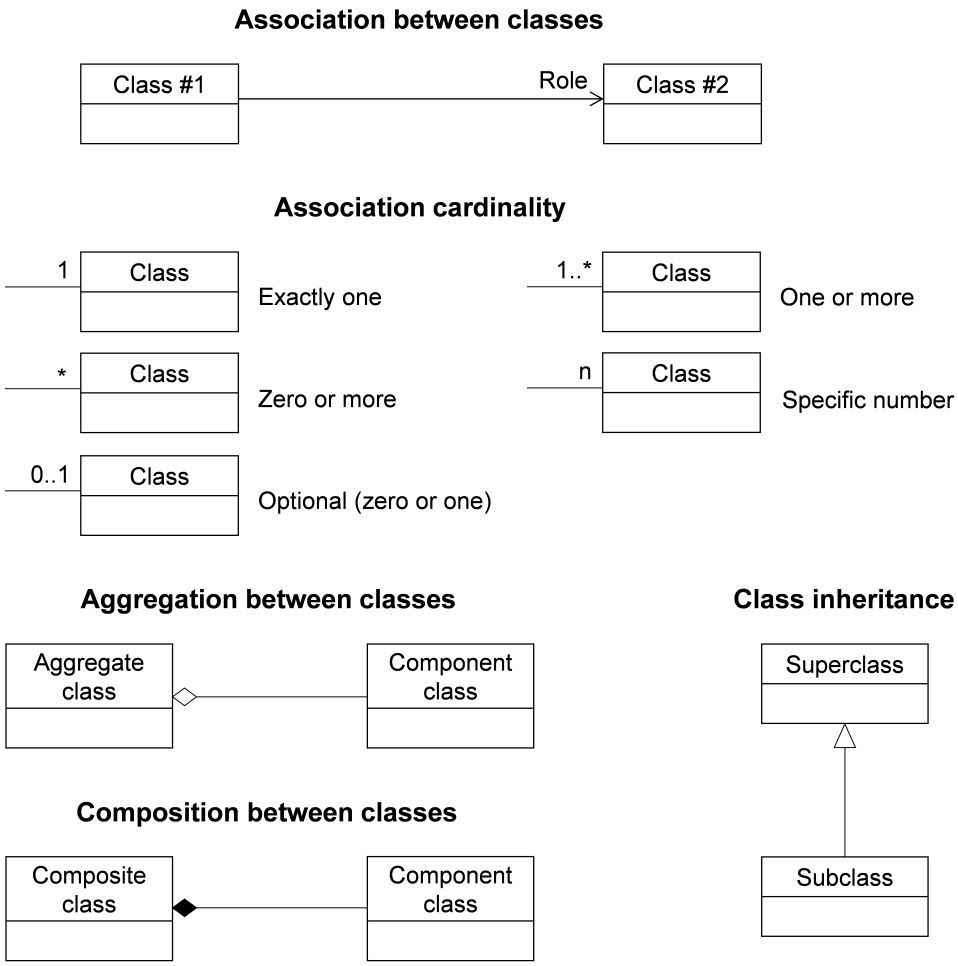

Figure 1 — UML notation

5

SPACE OBJECTS

5 SPACE OBJECTS

Introductory text

5.1. A Feature Model for Space Objects

Before we can describe Space Objects, 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.

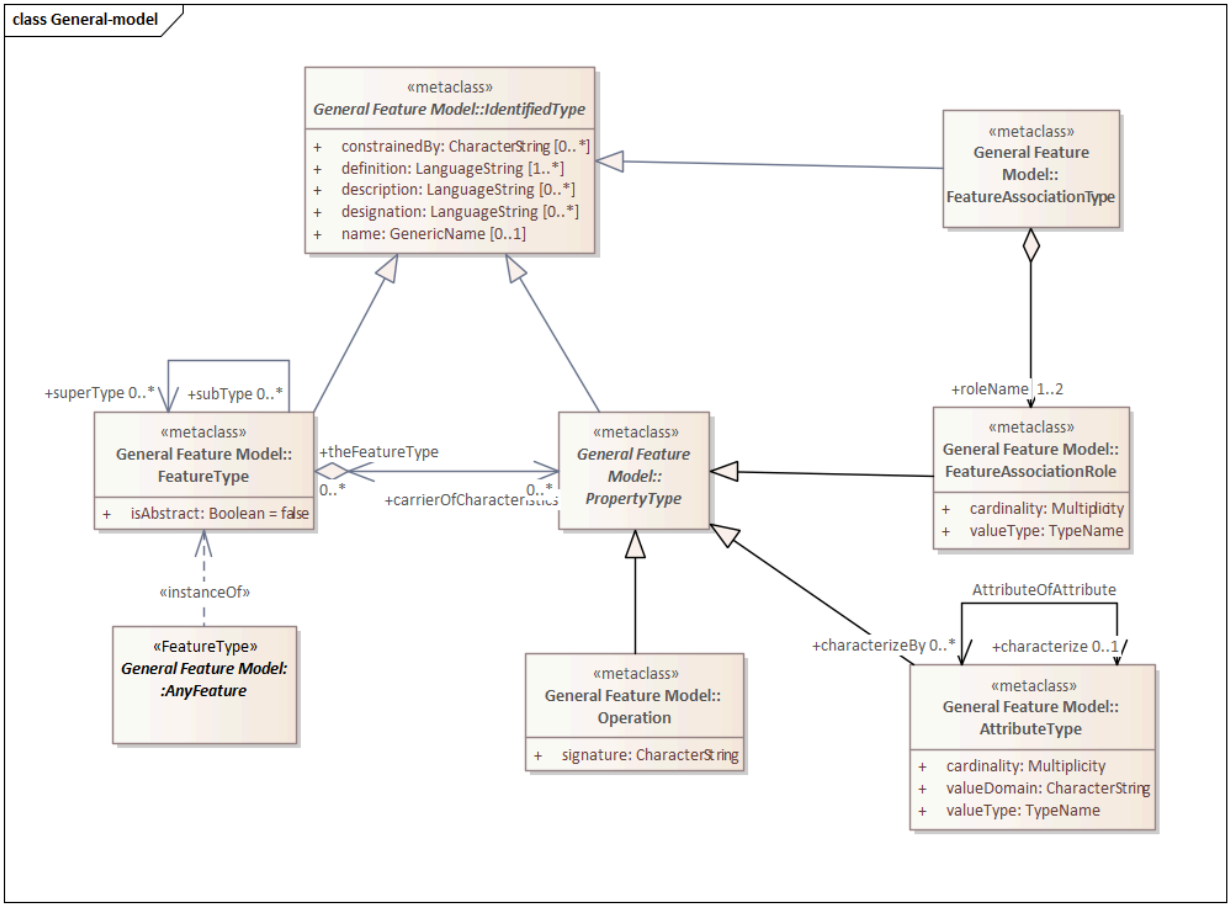

Figure 2 — General Feature Model

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

Conclusion: ISO 19109 provides a model for Real World Objects that is suitable for non-terrestrial use.

5.2. Geometry in 3 Dimensions

The applicable OGC/ISO standard for 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.

5.2.1. Features and 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)

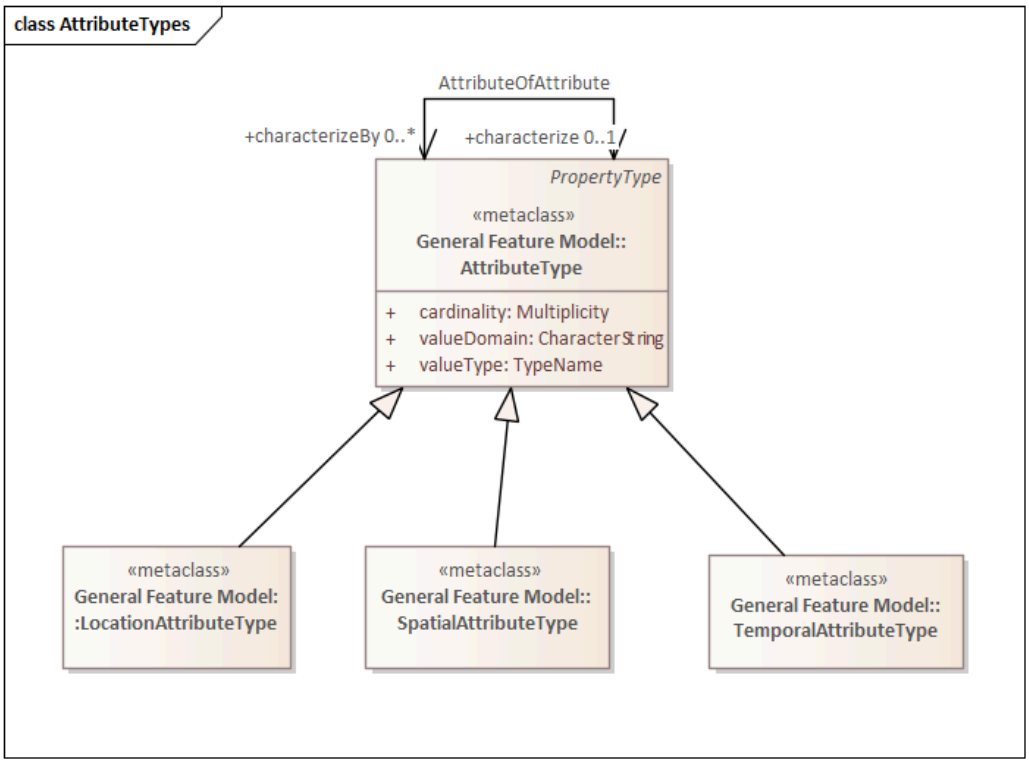

Figure 3 — General Feature Model Attribute Types

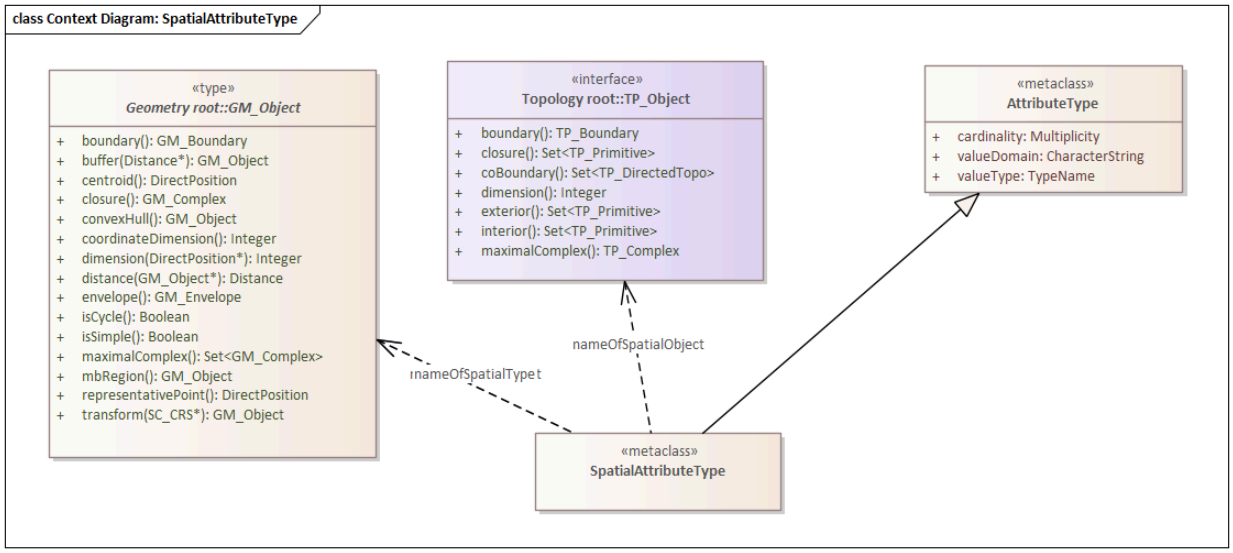

Figure 4 — General Feature Model Spatial Attribute Types

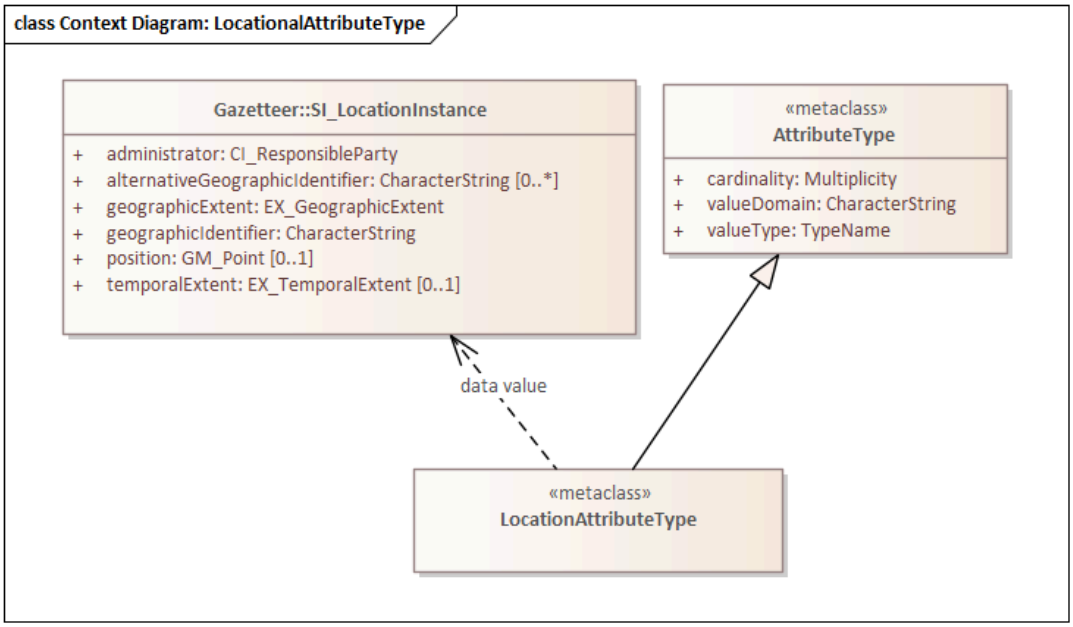

Figure 5 — General Feature Model Locational Attribute Types

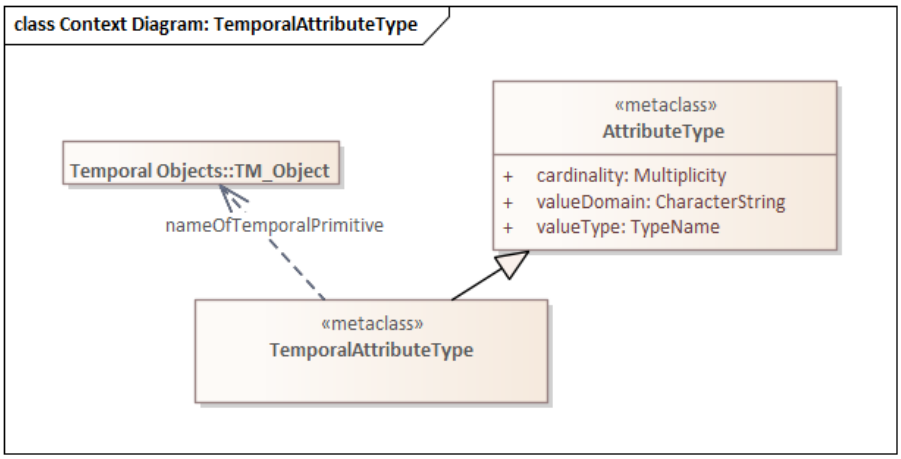

Figure 6 — General Feature Model Temporal Attribute Types

5.2.2. The Geometry Model

An important feature of the ISO Geometry Model is that it does not specify or assume a coordinate reference system. The SC_CRS class, defined in ISO 19111, is used to define a coordinate reference system. The “Coordinate Reference System” association is used to associate a GM_Object instance with the appropriate SC_CRS instance. Since all geometry classes are decended from GM_Object, any geometry object can have its own unique coordinate reference system.

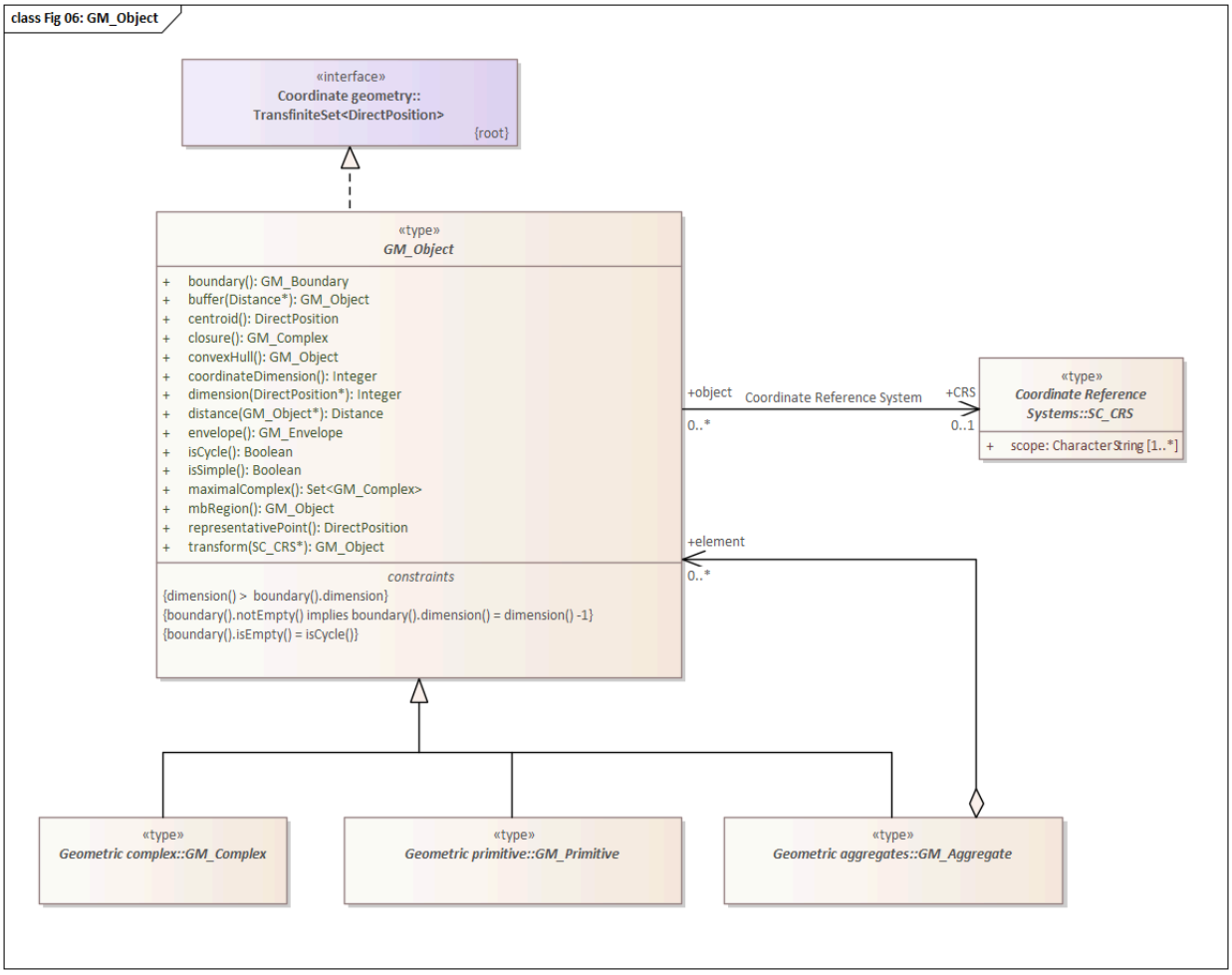

Figure 7 — GM_Object UML Model

While the ISO Geometry Model is very complex, at its core is the DirectPosition class. This is the fundamental specification of a position within a coordinate reference system. Its purpose is simply to hold the coordinates for a position within the coordinate reference system.

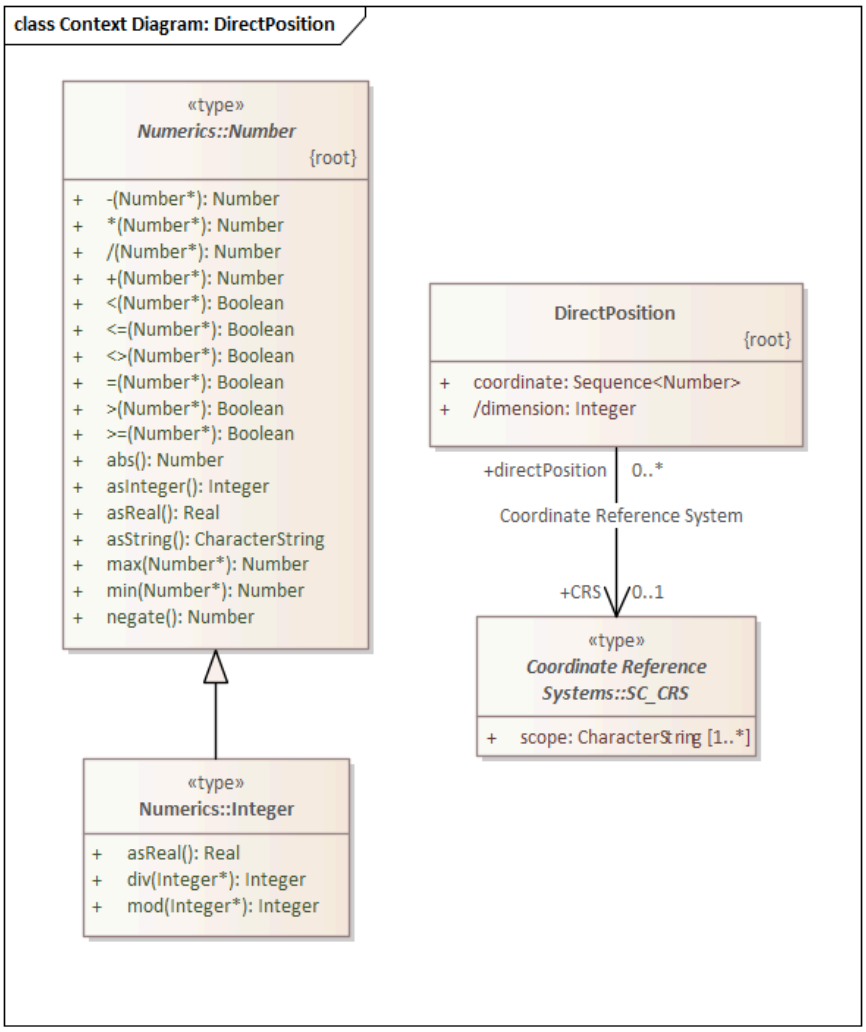

Figure 8 — Direct Position UML Model

The DirectPosition class contains two attributes, “dimension” and “coordinate”. The “coordinate” attribute is a sequence of numbers. Each number represents the location of the DirectPosition on a coordinate axis. There are no constraints on the number of numbers in the sequence. Therefore, a DirectPosition can represent an unlimited number of dimensions. The “dimension” attribute specifies the number of axis in the applicable coordinate reference system. This corresponds to the number of values in the “coordinate” sequence.

Like GM_Object, DirectPositions are associated with an SC_CRS through the cordinateReferenceSystem association. This association is typically not used since DirectPositions, as data types, will usually be included in larger objects (such as GM_Objects) that have their own references to SC_CRS. When this association is left NULL, the coordinate reference system of the DirectPosition instance will take on the value of the containing object’s SC_CRS.

One limitation of the DirectPosition class is that it does not support complex numbers. However, since DirectPosition does have a “coordinateReferencesystem” association with SC_CRS, it should be possible to model complex numbers in the CRS definition as as two orthagonal axis.

Conclusion: The ISO 19107 Geometry Model is suitable for geometries of three or more dimensions.

Issue: Support for coordinate values which are complex numbers needs to be validated.

5.2.3. Features in 3D

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 measured from the perspective 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 refrence system.

5.2.3.1. 3D Geometries

ISO 19107 makes a distinction between a geometric object and the surface which contains that object. One advantage of this approach is that there can be multiple surfaces associated with one object. For example, an island located in a lake would be represented by an interior surface (the island) of a polygon (the lake) bounded by the exterior surface (the shoreline).

ISO 19107 uses the GM_Object class to define an object and the GM_Boundary class to define a containing surface. Both GM_Object and GM_Boundary are defined as root level geometry classes. The association between GM_Object and GM_Boundary is achieved through the "boundary()" operation on the GM_Object class. This operation is inherited by all subclasses of GM_Object.

In the case of a 3D Feature, GM_Solid is the subclass of GM_Object while GM_SolidBoundary is the subclass of GM_Boundary. GM_Solid describes the volume while GM_SolidBoundary describes the shape.

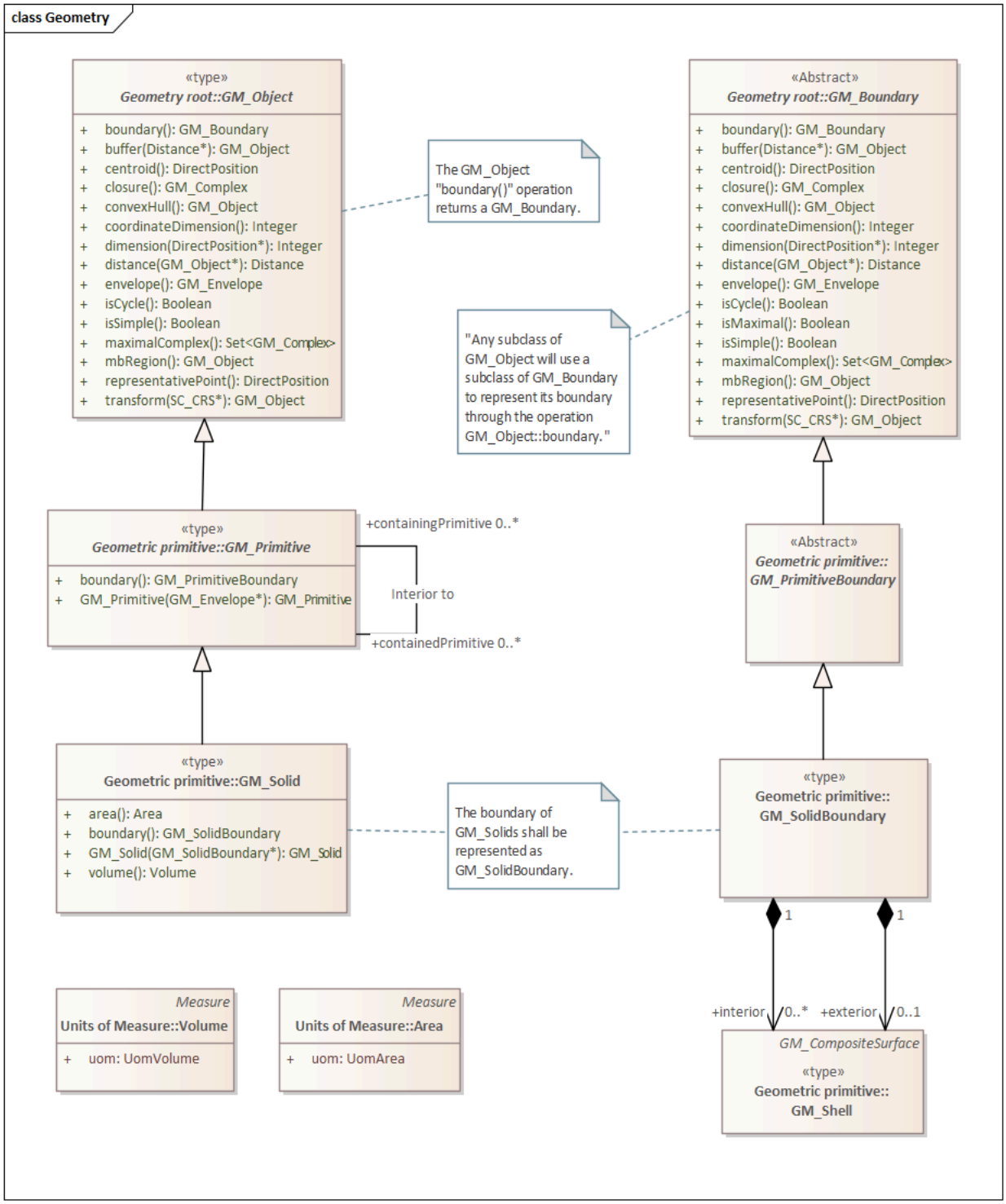

Figure 9 — 3D Geometry UML Model

5.2.3.2. Volumes

GM_Object is subclassed into GM_Primitive and then into GM_Solid. The “volume()” operation on GM_Solid returns the volume (defined in ISO 19103) of space contained within that GM_Solid. 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.

5.2.3.3. Shapes

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.

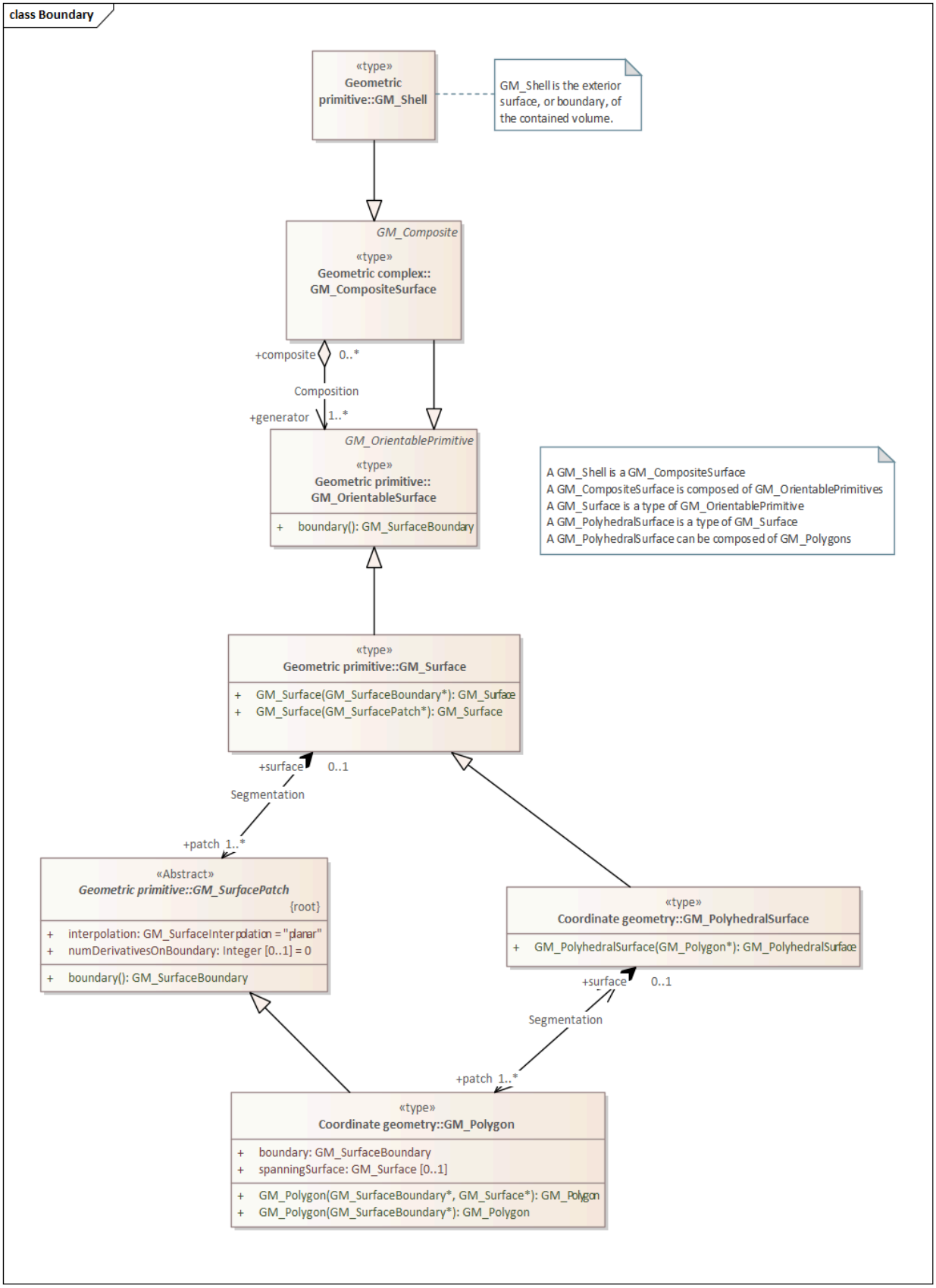

Figure 10 — 3D Boundaries UML Model

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 GM_PolyhedralSurface which is composed of GM_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)

5.2.3.3.1. 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 three-dimension solid, even though there is a hole in the bounding surface.

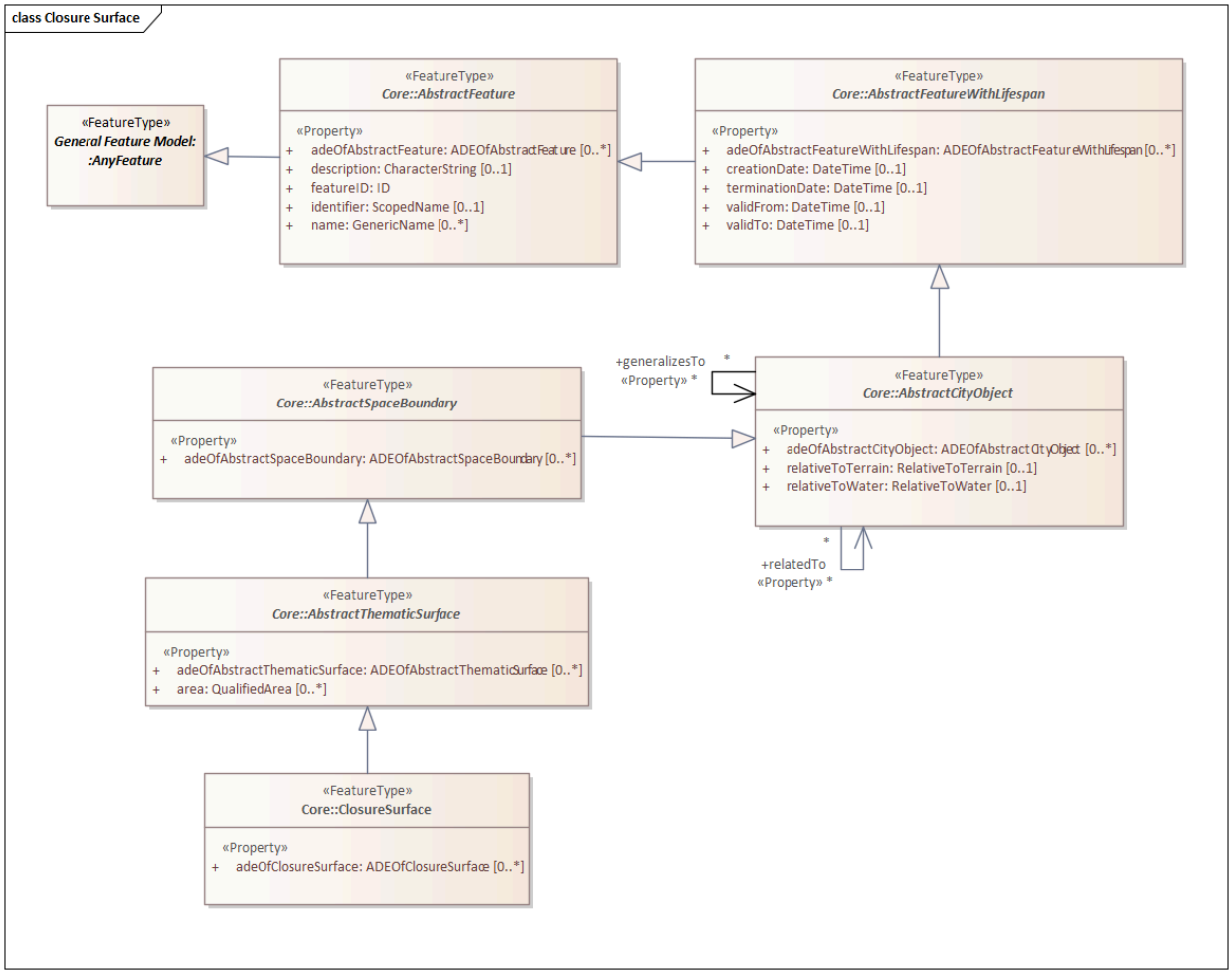

Figure 11 — Closure Surface UML Model

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.

5.2.3.3.2. 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

5.2.3.3.3. Point Clouds

Boundary surfaces can also be defined using 3D point clouds. This allows the spatial represention a bounding surface by a set of points located on that surface. In this way, the geometry of a Feature could, for instance, be modelled directly from the result of a mobile laser scanning campaign.

5.2.4. Conclusions and Future Work

The ISO 19107 Geometry Model appears to be suitable for representing complex, non-terrestrial objects of three or more dimensions. But will it work in reality?

An example of this Geometry Model applied to a practical application can be found in the CityGML family of standards. CityGML uses the 19107 geometry model to define buildings, the exterior spaces surounding the buildings, as well as interior spaces and even movable furniture. This should be sufficient for any complex object, whether on the surface of the Earth or in Space.

Conclusion: The ISO 19107 Geometry Model is sufficient to represent the geometry of complex space objects.

Two techniques have been described to represent the surface of a 3D object, point clouds and B-Rep. This list is certainly not exhausitve. According to ISO 19107, a GM_SolidBoundary object is composed of GM_Shell objects. But the Standard does not provide a decomposition of GM_Shell. So there is no way to specify the geometry underlying the shell.

Issue: Can and should we extend the GM_Shell class of ISO 19107 to address the underlying geometry of the surface of a 3D object?

The GM_Solid and GM_SolidBoundary classes are designed to represent a 3D space. It is not clear if they can also represent a 4D or even 5D space. While it's clear that the basic coordinate representation is independent of the number of dimensions, that may not be true of the more complex geometry constructs.

Issue: Can the GM_Solid and GM_SolidBoundary classes represent an n-dimensional solid?

5.3. 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.

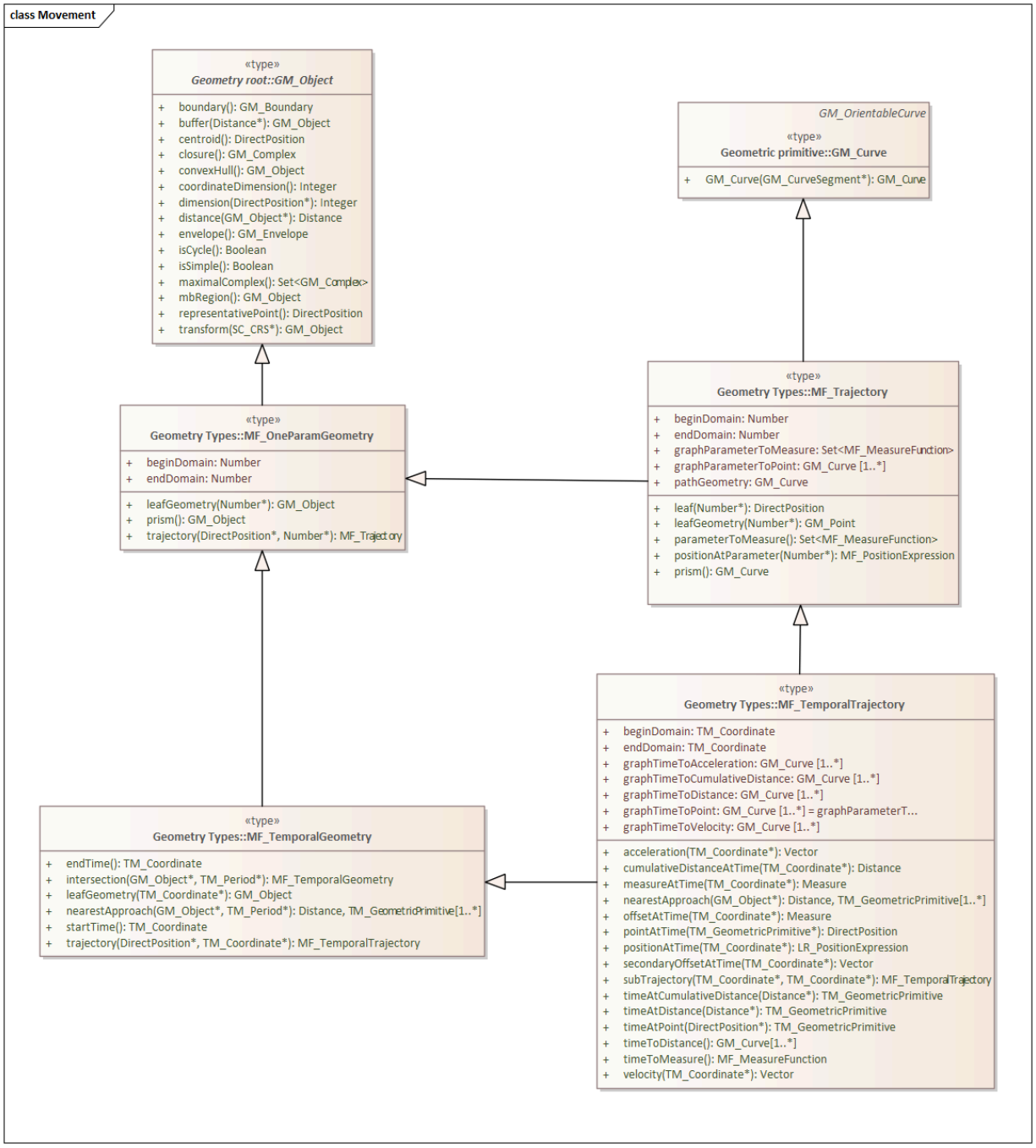

Figure 12 — Moving Features

A high level view of ISO 19141 is provided in Figure 12. The classes identified in this figure are described below. But first, a discussion of coordinate reference systems (CRS) is in-order.

5.3.1. Coordinate systems

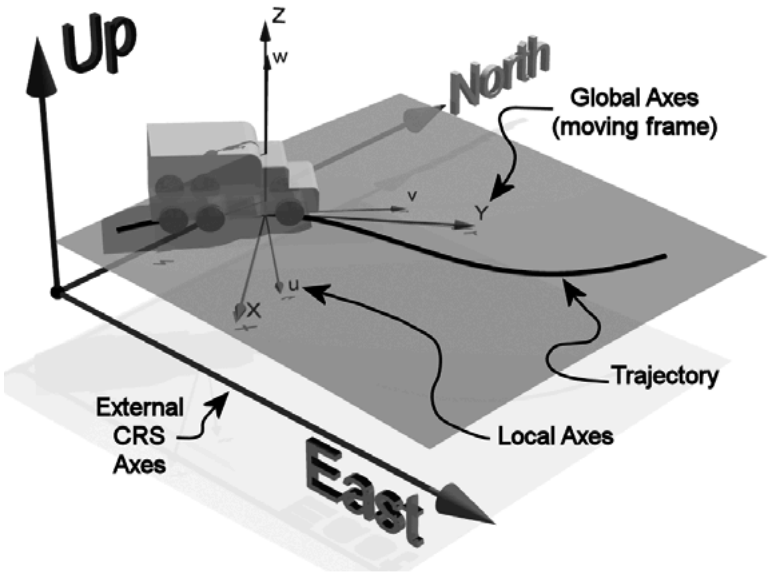

Figure 13 — External, Global, and Local CRS

Moving Features deal with three spatial coordinate systems as well at one temporal coordinate system. The spatial coordinate systems are referred to as the External, Global, and Local CRS (see Figure 13).

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.

5.3.1.1. External CRS

The External coordinate system is the coordinate system within which the Moving Feature exists. Typically this is an Earth-centric geographic CRS such as WGS 84.

5.3.1.2. Local CRS

The local coordinate system is internal to the Feature. This is usually a cartisian coordinate system with the origin at a prominent point in the Feature such as the center of mass. For rigid bodies, the local coordinate system is fixed over time. It does not change reqardless of any motion by the associated Feature.

This constraint does not hold for non-rigid bodies. This case will be discussed in the sections on articulated geometries and plastic geometries.

5.3.1.3. Global CRS

The Global coordinate system provides a transition between the External CRS and Local CRS. It is a moving CRS which follows the trajectory of the Moving Feature. As such, it provides the External CRS a time variant local reference system, while providing a time invariant context for the Local CRS.

The origin of the Global CRS is the location of the Moving Feature on the trajectory curve at a specific time. From the perspective of the External CRS, the origin translates as a funcion of time. From the perspective of the Internal CRS, however, the origin of the Global CRS is static.

The axis of the Global CRS can be defined using two techniques.

The first approach treats the Moving Feature as a black box viewed by an external observer. From this perspective, the Global CRS is defined in terms of the trajectory alone. No knowledge of the properties or even the shape of the Moving Feature are required. This CRS starts by defining x and y as two orthagonal axis which define a plane tangential to the Trajectory curve at the origin. Positive x is in the direction of motion. The positive y axis is perpendicular to the x axis and forms either a right or left handed CRS. The z axis is perpendicular to the tangent plane. Positive z can be either up or down. The result is a cartesian refernce system which is always tangential to the trajectory of the Moving Feature.

The second approach defines the Global CRS in terms of motion properties of the Moving Feature. The x axis, for example, could be defined as the heading of the Moving Feature. This is the direction the Feature is pointing, but not necessarily the direction it is moving. The y and z axis can be defined using similar properties (for example; pitch, yaw, and roll).

In general, the black box approach is appropriate for ballistic Moving Features. Cases where the Feature is not anticipated to take any actions which would modify the trajectory. The motion properties approach is appropriate for navigated Moving Features. Cases where a Feature is expected to take actions which would modify the trajectory.

5.3.1.4. 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.

5.3.2. 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 be 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 one member with the name “datetimes”.
- The value of the “datetimes” member is a JSON array.
- Each element in the “datetimes” array SHALL be an instant object.
- An instant object SHALL be a JSON string that represents a timestamp encoded in 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 members of the `"datetimes"` array 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] } },

Figure 14

5.3.3. Geometry

5.3.3.1. MF_OneParameterGeometry

We start our discussion of Moving Feature geometries 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.

NOTE: verify the following definition and clarify the symbology. It does not appear to render correctly.

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.

5.3.3.2. 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.

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

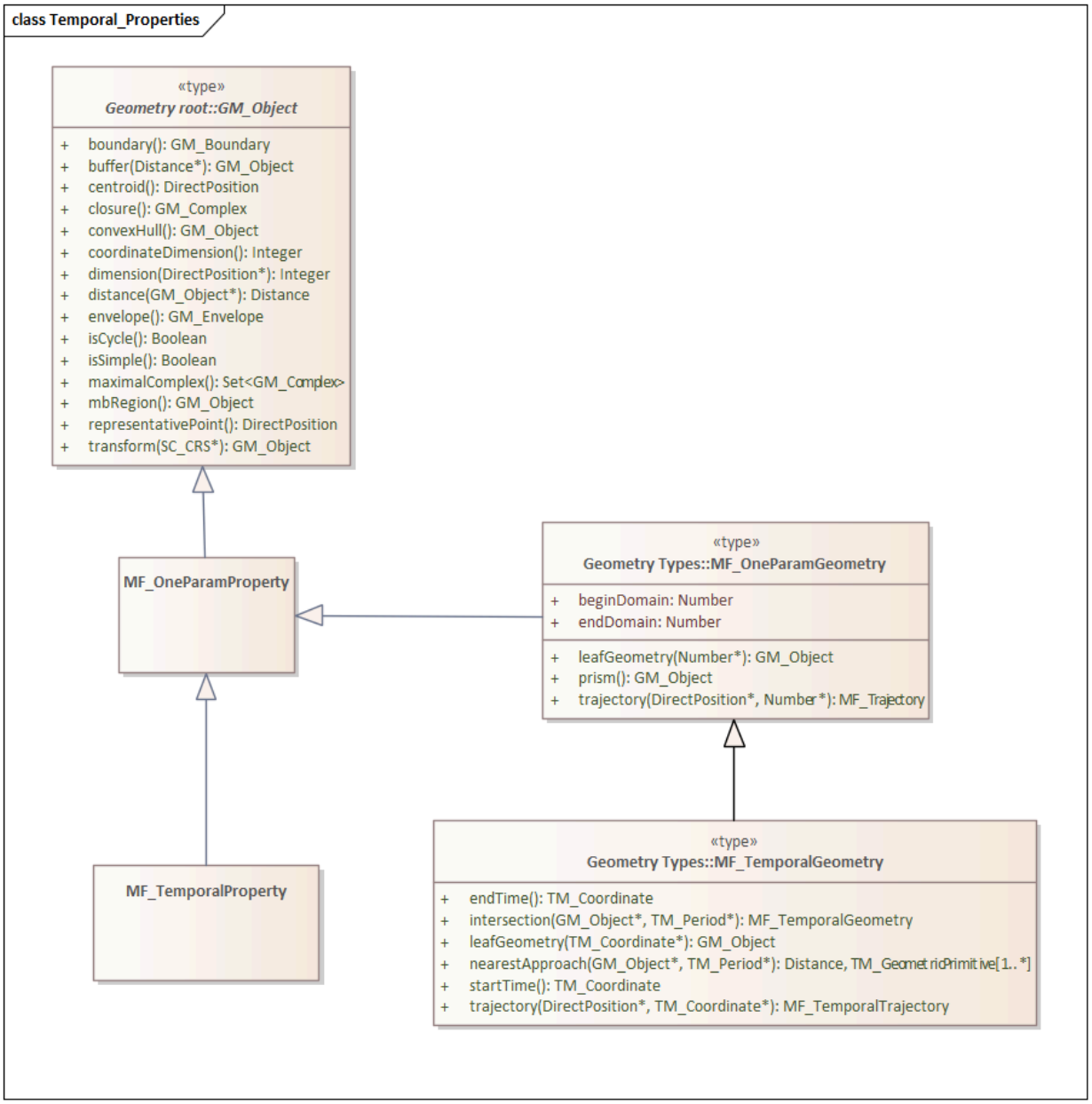

Figure 15 — Temporal Properties

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.

5.3.4. Location

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

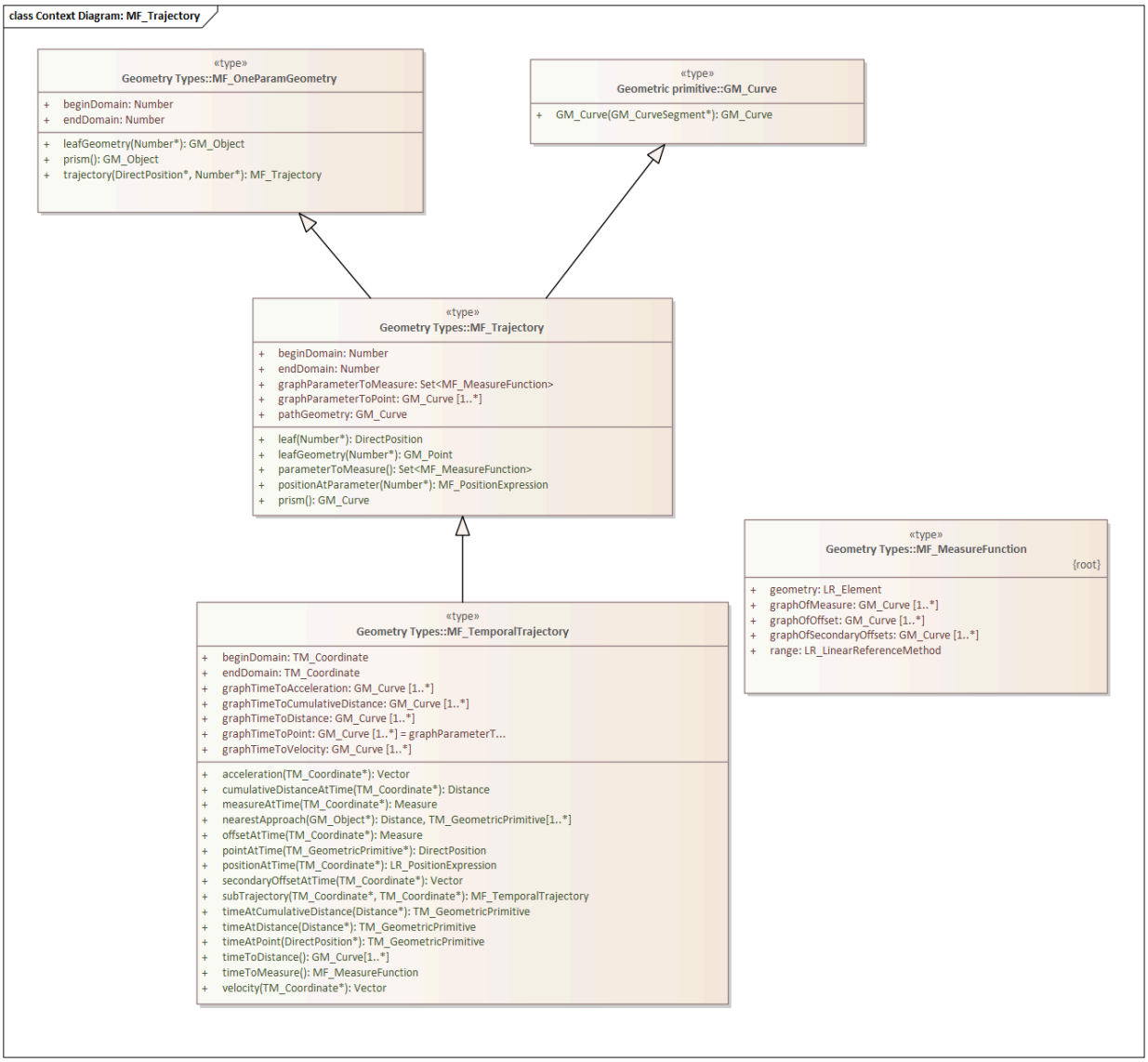

Figure 16 — Trajectory

A MF_Trajectory is a curve (GM_Curve). It represents every postion that the Feature has occupied during it's 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 this Moving Feature possesses at the time (TM_Coordinate) specified by the input parameter. This is the shape of the Moving Feature expressed in the Local CRS. Since Trajectories only convey location, only GM_Point geometries are supported.

5.3.5. Orientation

5.3.5.1. 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, orientaion, and shape) of the Feature as a function of time.

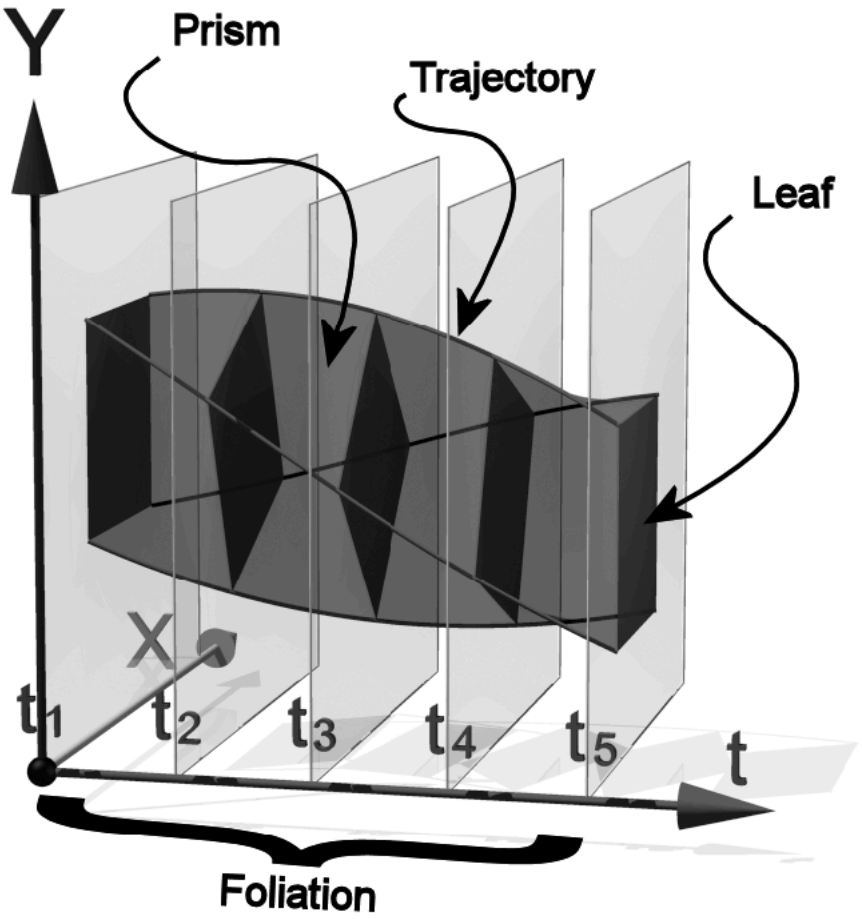

Figure 17 — Foliation

The key concepts in the Prism model are:

Leaf: A leaf is the geometry of the Moving Feature at time (tn).

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

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.

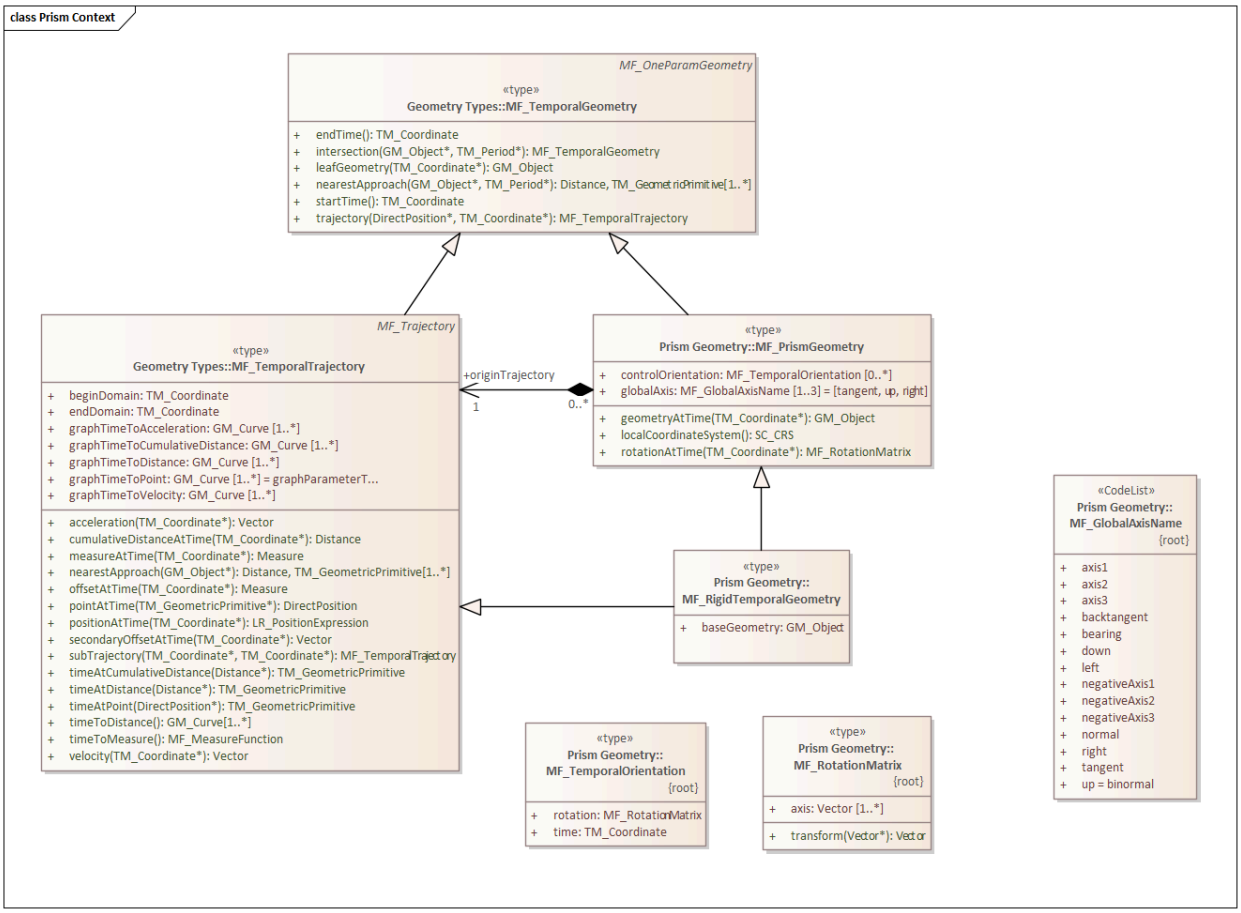

Figure 18 — Prism Context

A MF_PrismGeometry class has the following characteristics.

The association role “originTrajectory” associates a Temporal Trajectory with a Prism geometry. For any TM_Position:

1. the associated Temporal Trajectory provides the location of the Moving Feature in the Global CRS.
2. this location serves as the origin of the Local CRS.
3. the prism geometry is defined in that Local CRS.

The localCoordinateSystem() operation returns a SC_CRS for the design coordinate reference system in which the moving feature’s shape is defined. This is usually the same as the local coordinate system.

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 funtion of time (tn).

5.3.6. 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.

5.4. 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 one (1) to six (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 may change as the relative position of the elements change.

Two existing standards address the problem of Articulated Geometries. The MISB Staging System was developed for articulated Motion Imagery capture systems. GeoPOSE also addresses Motion Imagery, but it is more focused on Augmented and Virtual reality.

It is not likely that either standard provides a complete solution. Their relative merits and deficiencies are discussed in the `Discussion` section as well as proposals for work to be done.

5.4.1. MISB Staging System

The Motion Imagery Standards Board (MISB) develops standards for motion imagery, audio, associated metadata, and their related systems. Exploitation of MISB data requires its precise positioning in space and time. Therefore, a number of MISB standards deal with the derivation of ground coordinates from sensor coordinates.

Motion Imagery systems are typically complex structures consisting of multiple components each capable of independent motion. Measurements of these motions are in a local coordinate reference system. Conversion of a sensor coordinate requires a series of transformations, starting from the detector and working back through the attached components to a core, geolocated component.

Further complicating maters, each component is capable of movement. The transformations are not fixed. They must be calculated from the relative position of each component at the time that the detection was made. This leads to a requirement that for every observation, the relative position and orientation of each component of the collecting system must be captured as well.

The MISB 1906 Motion Imagery Metadata (MIMD): Staging System standard addresses this requirement. This standard captures the locations, orientations, and kinemantics (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.

A UML model for the MISB Staging System is provided in Figure 19.

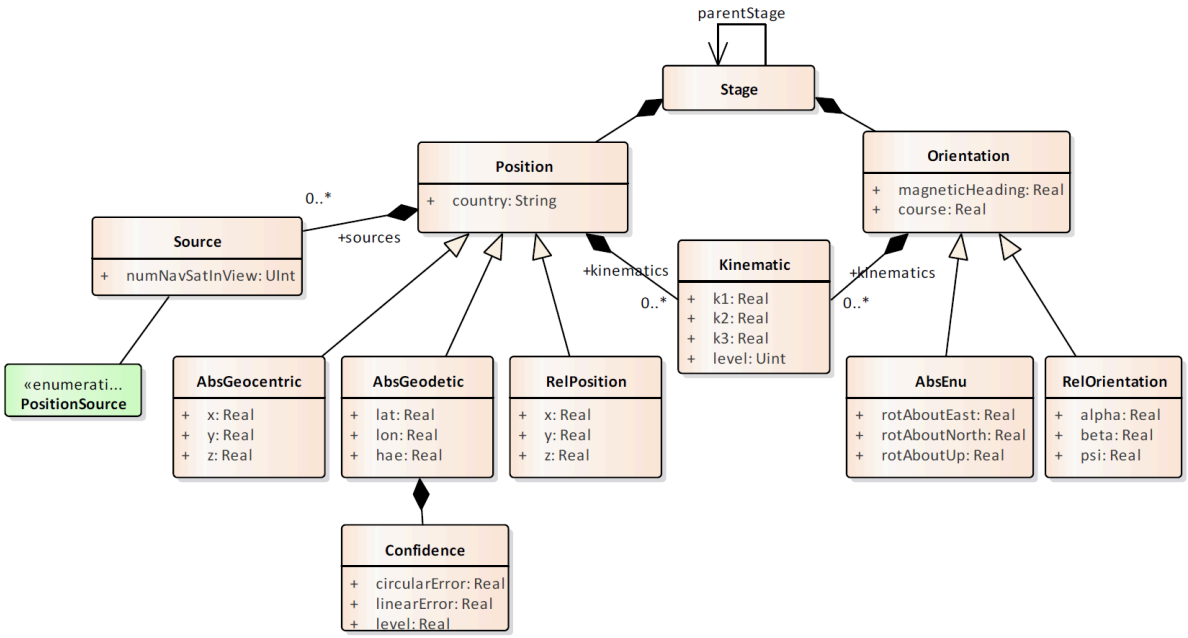

Figure 19 — MISB Staging System UML Model

This model only addresses a single stage. Non-trivial systems will require multiple stages. These stanges are assembled using the `parentStage` association of the `Stage` class. The following three concepts guide the construction of multiple-stage models:

- **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 (AbsGeocentric or AbsGeodetic).

An example articulated motion imagery system is illustrated in Figure 20. The stages which describe this system are described in Table 1.

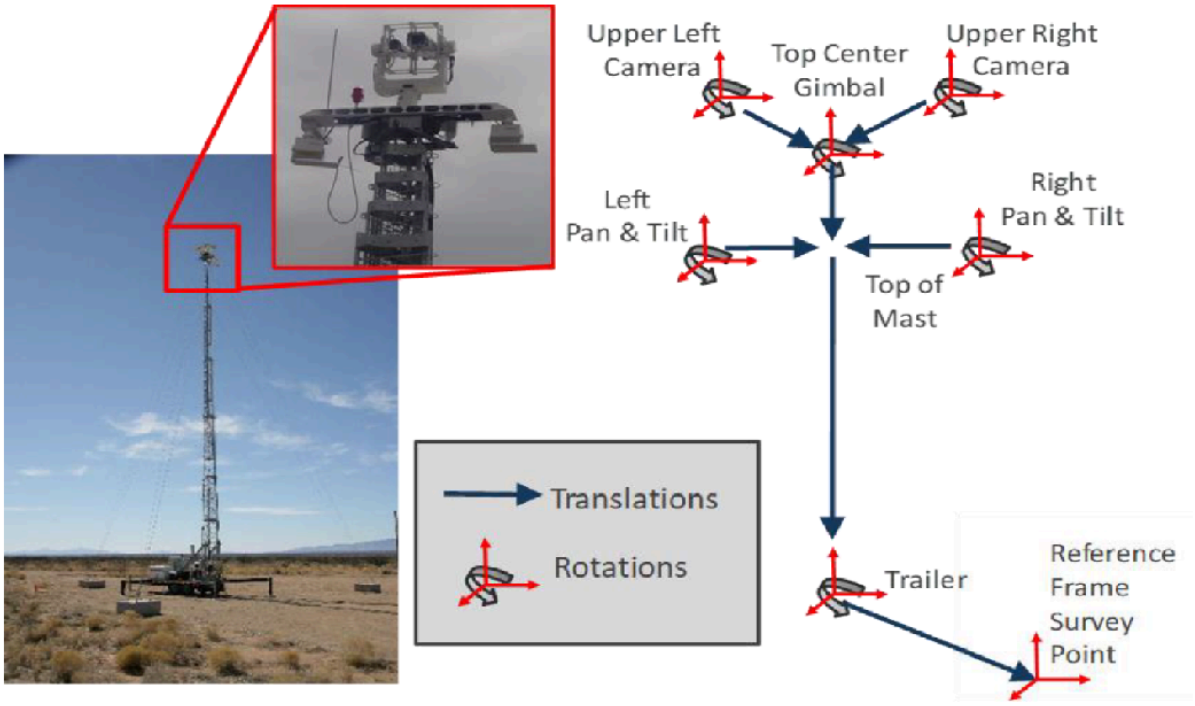

Figure 20 — MISB Staging System Example

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

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) axis 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. 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 ——

5.4.2. GeoPOSE

GeoPOSE is a proposed OGC standard which provides a general purpose alternative to the MISB Staging System. This standard deals with the location and orientation of real or virtual geometric objects (Poses) and the aggregation of Poses into more complex structures.

5.4.2.1. Core

A UML model for the core GeoPOSE concepts is provided in Figure 21

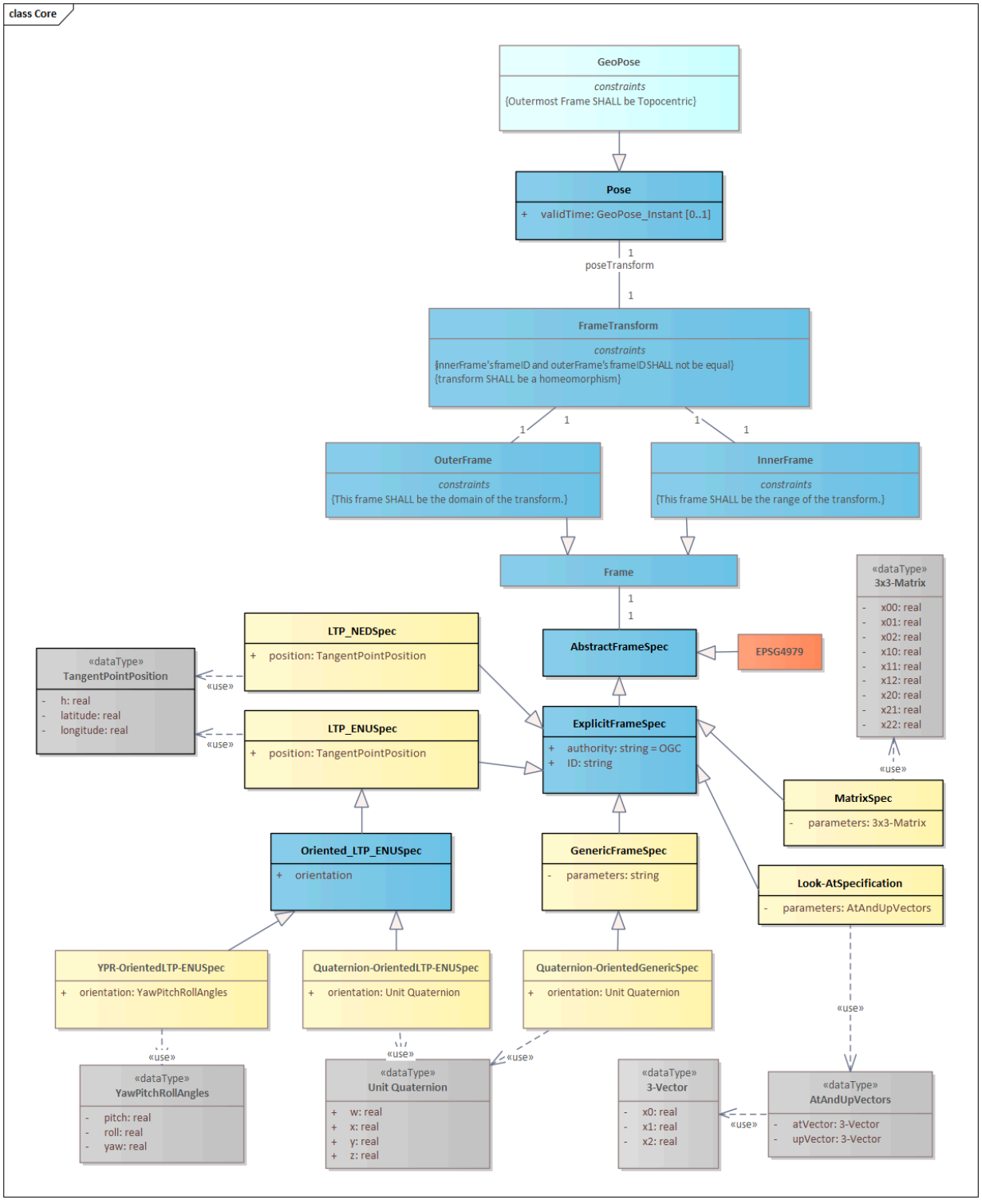

Figure 21 — GeoPOSE Core

The key element in this model is the `FrameTransform` class. This class expresses a transform between a pair of Reference Frames, Outer and Inner, each defined by a Frame Specification.

The Frame Transform is a representation of the transformation taking an Outer Frame coordinate system to an Inner Frame coordinate system. GeoPose v 1.0 supports

transformations involving translation and rotation. The intention is to match the usual concept of a pose as a position and orientation.

Outer and Inner Frames are subclasses of the the Frame Class. The Frame class provides a standard means of describing transforms for several common coordinate reference systems. Subclasses of the Frame class also include a model for generic reference frames.

5.4.2.2. Time

The GeoPOSE time model is rather simple. It does away with calender and restricts time positions to milliseconds of UNIX Time.

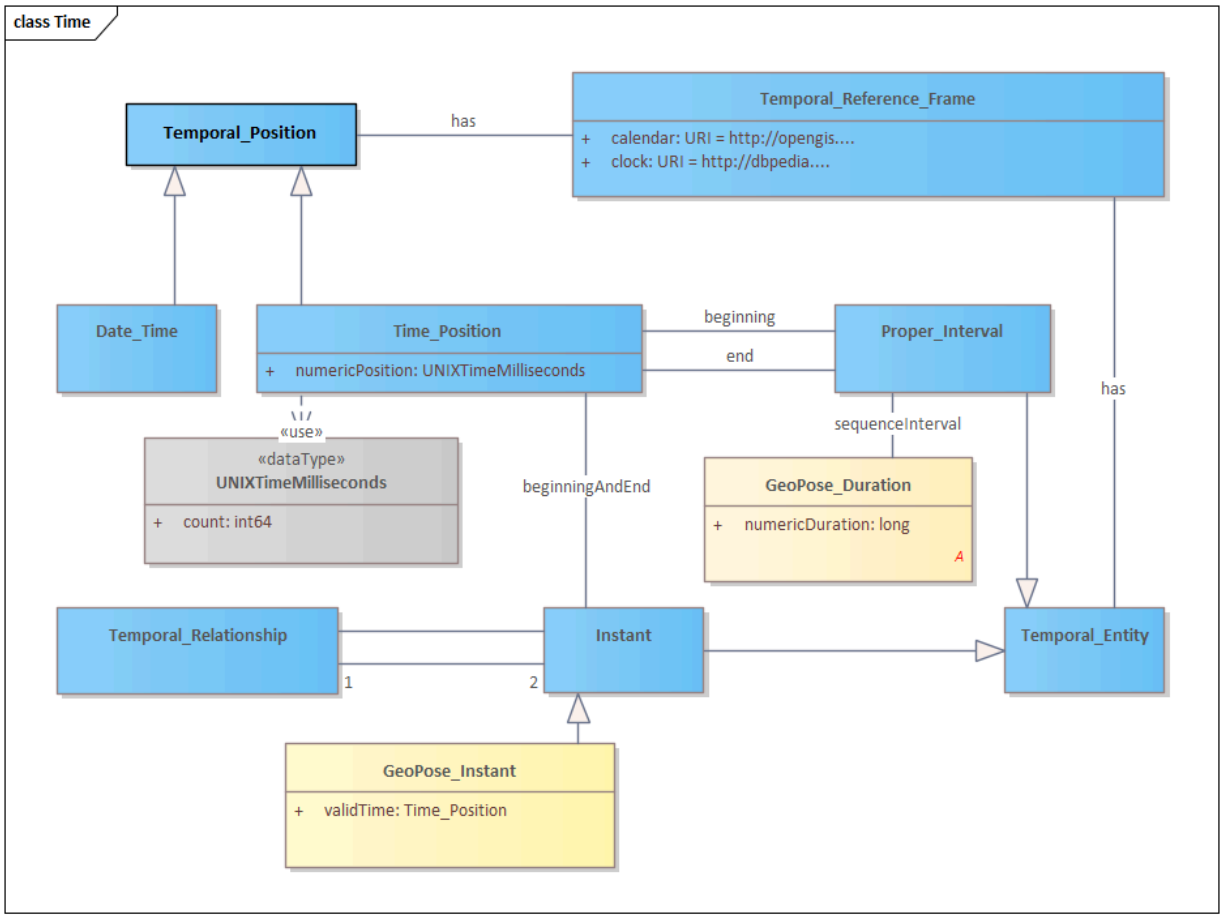

Figure 22 — GeoPOSE Time

The most important classes are GeoPose_Instant (a single point in time) and GeoPose_Duration (a period of time).

5.4.2.3. Sequence

The sequence logical model defines methods for packaging GeoPose transformations. It addresses the need to integrate multiple GeoPoses which share the same Outer Frame and posess a time-dependent changing Inner Frame.

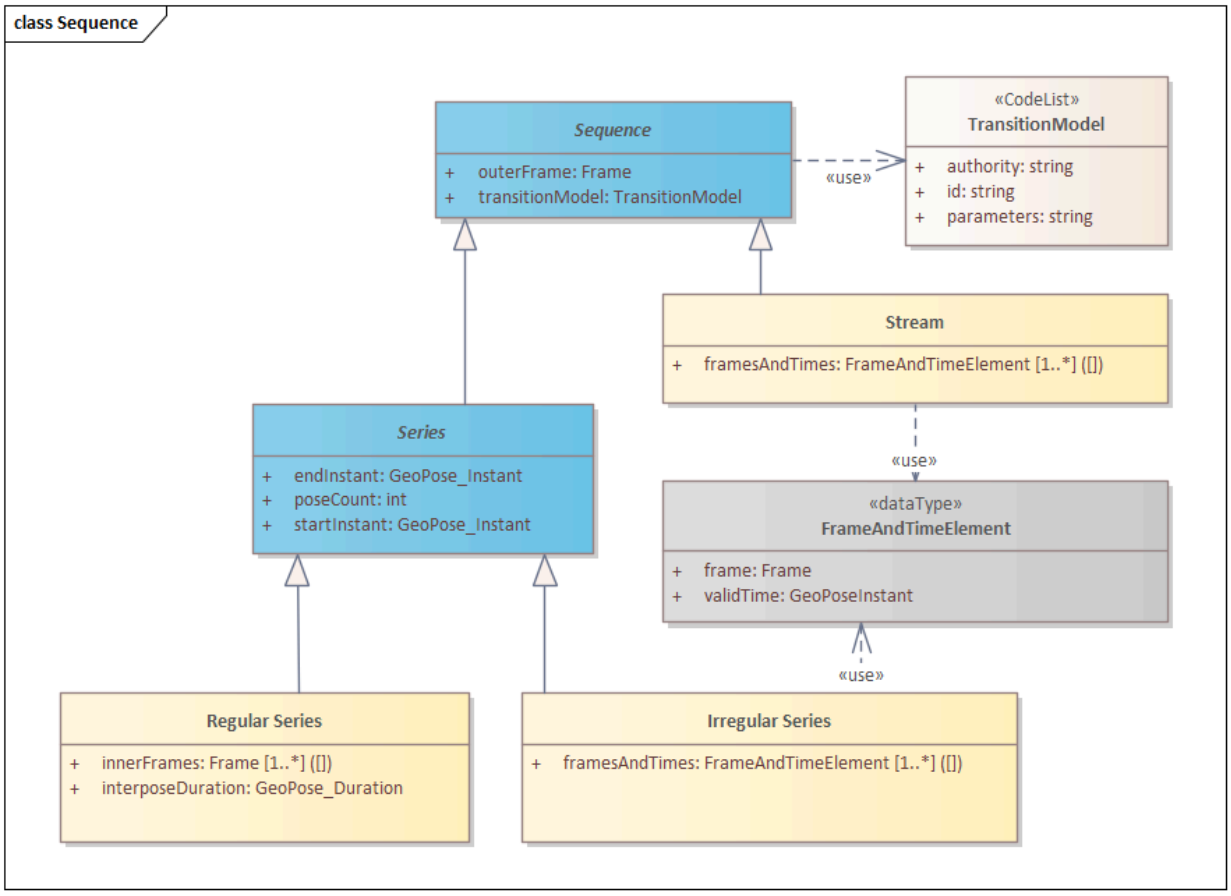

Figure 23 — GeoPOSE Sequence

The GeoPOSE Standard provides three models for organizing a collection of Inner Frames.

- Stream: the Inner Frame definition (Frame) and an associated time stamp are delivered sequentially.
- Regular Series: the Inner Frame definitions are delivered as a sequence, separated by a fixed time interval.
- Irregular Series: the Inner Frame definitions and associated time stamps are delivered as a collection. There is no explicit spatial or temporal order to the frames.

5.4.3. Discussions

Both models express some common concepts:

1. There is an “anchor” node which ties the local coordinates to an absolute (external) CRS.
2. Coordinates in one Local coordinate system can be transformed to another through standardized transformations.
3. Local coordinate reference systems are not pre-defined. Sufficient information is defined to describe the transformation between any two arbitrary local coordinate systems.

5.5. Plastic 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.

5.6. Mass Properties

A space object does not have a medium to hold it in place. As a result, these objects have six degrees of freedom. They can move laterally along the x, y, and z axis. They can also rotate around the x, y, and z axis. This freedom of motion means that the movement of an object is dependent, not just on the forces applied, but where these forces are applied.

When a force is applied to a 3D object in free flight, the resulting motion depends on the mass of the object and how that mass is distributed throughout that object. These are the mass properties of the object.

5.6.1. Linear motion

Linear motion occurs when a force (or forces) are applied to an object in such as way that there is a change in the objects velocity but no change in rotation.

An analysis of linear motion starts with the mass of the object. The observed mass of an object is usually the same for all observers. Therefore, we should treat mass as a physical property that should be present in all 3D Features.

The mass of an object is distributed across the object. Yet we often want to deal with that mass as a point mass, a mass of quantity *M* located at coordinates (x,y,z). The position of this point

mass is called the `Center of mass`. The center of mass, however, is not an arbitrary point. This is the one location where the distributed mass of the object ballances out. Where the moments of inertia sum to zero. This allows the object to be treated as a single point with a mass of *M* and located at the `center of mass` (x,y,z). Both of these parameters must be present and defined in the local coordinate reference system.

NOTE: If the object is not a rigid body, then the center of mass may move as the object changes configuration.

Newtons First Law states that “Every body continues in its state of rest, or of uniform motion in a straight line, unless it is compelled to change that state by forces impressed upon it.” In other words, the velocity (speed and direction) of an object will not change unless you apply a force to the object. This property is known as inertia.

Inertia is governed by the formula F = M * A. Where F = Force, A = Acceleration, and M = Mass. The amount of force required to change the velocity by amount A is porportional to the mass of the object. While inertial is important, it is a computed value and not a property of the Feature.

When you use a force to accelerate an object that object gains momentum. Linear momentum is gained (or lost) when the force is applied through the center of mass. The value is the product of the rest mass and the accelerated velocity of the object. Since velocity is measured relative to the observer, this is not a property of the space object. Rather it is a value that must be computed by the observing system.

5.6.2. Rotational motion:

Linear motion is motion resulting when a force is applied through the center of mass of an object. Reality is rarely that accomodating. If the force is applied to a point even a little off the center or mass, the object will acquire both linear and rotational movement. This section explores the rotational movement.

Rotation of an object is usually evaluated in terms of three axis of rotation. These axis often align with the three axis of the local coordinate reference system. But not always. For example, satellites are often spun in order to stabilize them. This “principal axis of rotation” is selected based on the performance parameters of the satellite. The local coordinate system is not a governing factor.

The rate of rotation around each axis is the `angular velocity`. This value is typically measured in radians or degrees per second. Angular velocity is a physical property of the object so it is measured relative to the local coordinate system. While it can be measured by an observer, values measured and reported by the object would be more reliable.

Rotational motion must accomodate not only the mass of the object, but how that mass is distributed. This distribution of mass is represented by the moment of inertia. There is one moment for each axis of rotation. The moment of inertia is calculated by summing up all of the products of the point masses that make up the object multiplied by their distance from the rotaional axis. These mass and distance values are all measured in the local coordinate reference system. They are properties of the object and are not usually measured remotely.

Once a object starts to rotate, it has angular momentum. This is the product of the moment of inertia around one axis of rotation and the associated angular velocity. Since the components of angular momentum are properties of the object, the angular momentum is also a property of the object. As such, it is defined relative to the local reference system.

Like linear momentum, angular momentum is conserved. It cannot be changed except through the application of a force.

For example, consider a spacecraft with solar panels.

When launched, the panels are retracted close to the center of mass. Once free of the boosters, the spacecraft will be spun. This gives it angular momentum. The solar panels are then deployed. As the panels are deployed, their mass moves away from the axis of rotation. This increases the moment of inertia of the spacecraft. Since angular momentum is conserved, then the angular velocity must decrease.

6

TIME

6 TIME

6.1. Clocks and Timers

6.1.1. 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 nanosecond counter.

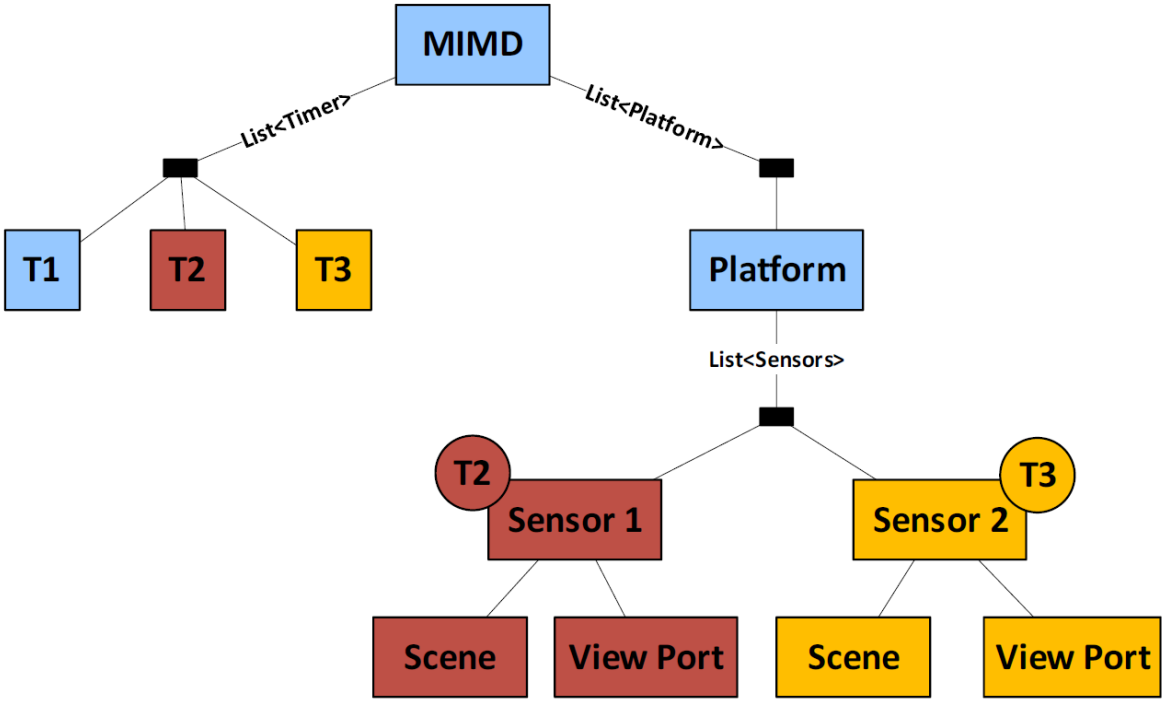

Figure 24 — MISB Timer Example

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.

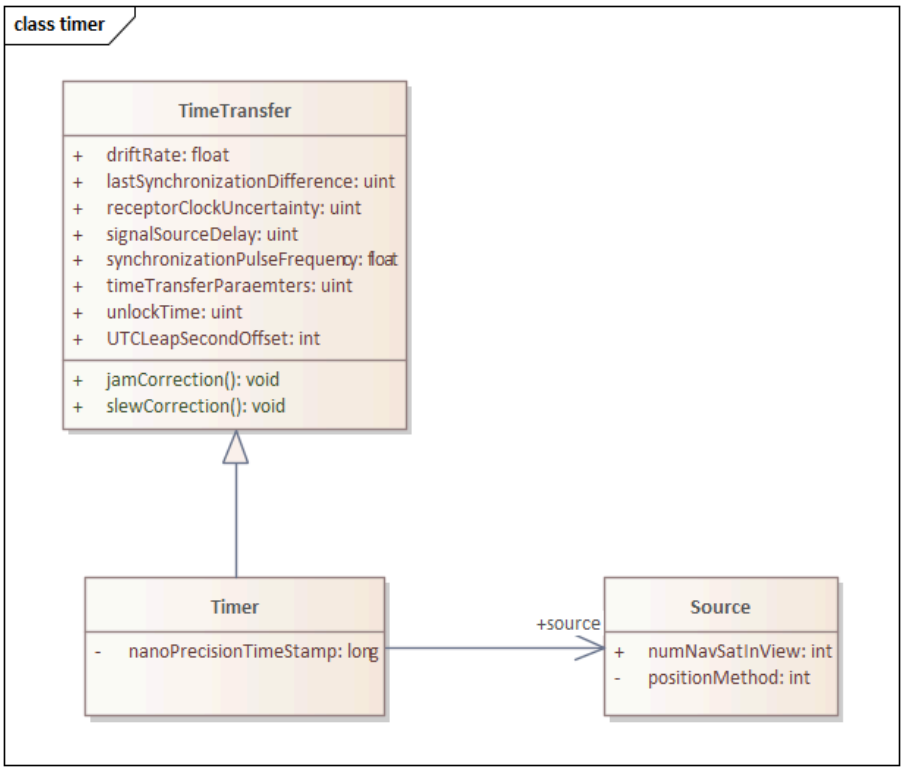

Figure 25 — MISB Timer Synchronization

figure — illustrates the Timer architecture. A Timer is a physical device which provides a time stamp with up to nanosecond precision. The timer is associated with a “Source”. The Source specifies how the timer is initallized and how any subsequent corrections are applied. Finally, the TimeTransfer pack defines the metadata available to detect and apply any corrections to the time stamp.

The MISB timer concept is specific to the platforms and collection techniques used for Motion Imagery. These can be extended to provide a more general solution.

6.1.2. Network Time Protocol

Do you really know what time it is?

A standard technique for synchronizing clocks is the IETF RFC 5905 Network Time Protocol (NTP).

NTP is a client-server protocol. A client requests the time from a Time Server, and the server provides a response with the current time. NTP provides a precision up to 2^{-64} seconds. However, precision is not accuracy. It takes time to process the NTP packets. It also takes time for the packets to traverse the network. So the accuracy of NTP can be as low as 0.01 seconds.

NTP addresses these sources of error by using a round-trip protocol. The initial request contains one timestamp – the time of the request packet transmission. The response adds two more

timestamps; the time that the request was received and the time that the response packet was transmitted. The final timestamp is the time of the response packet reception. So we have:

t_0 – Request packet transmission timestamp

t_1 – Request packet reception timestamp

t_2 – Response packet transmission timestamp

t_3 – Response packet reception timestamp

Note that the timestamps must be created as low in the protocol stack as possible to avoid additional CPU processing delays.

The difference between the client and server clocks is measured twice. First for the request packet ($t_1 - t_0$) and then for the response packet ($t_3 - t_2$). The total clock offset (θ) is the average of these two measures. Mathematically defined by the formula:

$$\Theta = ((t_1 - t_0) + (t_2 - t_3)) / 2$$

The round-trip delay (δ) is the time spent in transit between the transmission of a request and reception of the response. Mathematically defined by the formula:

$$\delta = (t_3 - t_0) - (t_2 - t_1)$$

The term $t_3 - t_0$ represents the total delay while the term $t_2 - t_1$ represents delay at the server. Since the processing time of an NTP packet should consistent, the second term removes this delay from the measurement. The result is the delay due to network latency, the most dynamic part of the problem.

Given these concepts, the NTP protocol can be viewed as follows:

1. Issue an NTP request
2. Receive the response
3. Calculate Θ
4. Calculate δ
5. IF either Θ or δ is a statistical outlier, GOTO 1
6. IF δ is symmetrical (request and response delays are equal) GOTO 8
7. Adjust the local clock in the direction of Θ (incremental corrections only)
8. Wait a bit
9. GOTO 1

This approach assumes a single time server. Additional accuracy can be achieved by using multiple time servers and adjusting to the weighted average of their respective offsets.

The NTP has one source of systematic error. If the Time Servers are not accurate, then their clients will not be accurate either. This problem is addressed through an infrastructure of network time servers. At the top (stratum 0) servers are high-precision timekeeping devices such as atomic clocks. Next are the stratum 1 servers. These are servers whose system time is synchronized to within a few microseconds of their **attached** stratum 0 devices. Stratum 2 servers are synchronized over a network to stratum 1 servers. Stratum 3 servers are synchronized over a network to stratum 2 servers. This pattern can be replicated down to stratum 15. Lower stratums naturally lead to less accuracy.

Note: The time server includes its stratum level in the NTP response packet.

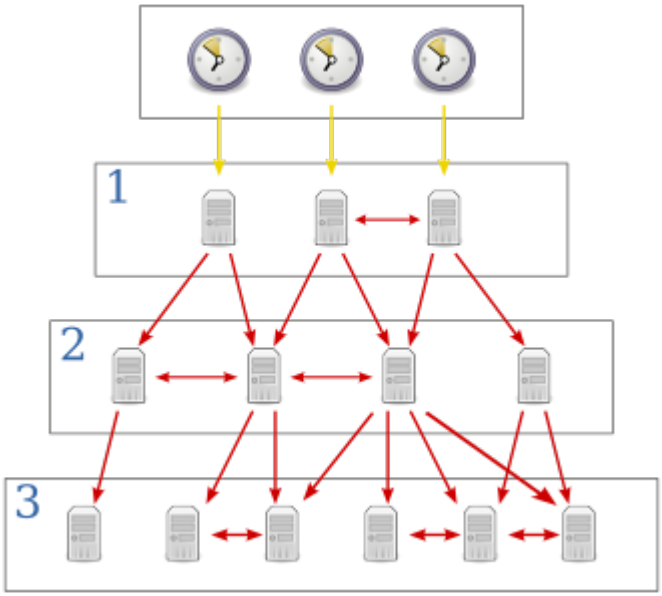

Figure 26 — Network Time Protocol

6.1.3. Electronic Distance Measurement

Electronic Distance Measurement (EDM) devices operate by sending a laser pulse to a reflective target and measuring the time until the reflection is received back at the instrument. Distance can be calculated using the following formula:

$\rho = c * (\Delta t / 2)$

Where:

ρ = the measured distance<

c = the speed of light

Δt = the interval between emission of the laser pulse and detection of its reflection

Figure 27 illustrates an EDM measurement. The blue sine wave indicates the emitted light. The Red sine wave indicates the reflection. The value P is the distance between the EDM and the reflector.

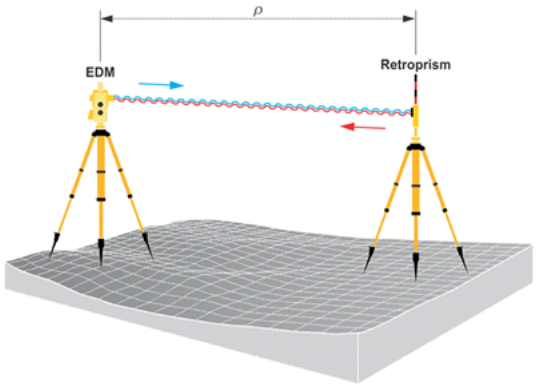

Figure 27 — Electronic Distance Measurement

Source: GPS for Land Surveyors

There is an issue though. With the speed of light at $3.0X10^8$ meters per second, a six-nanosecond error in the time measurement would equal a 1-meter error in distance. This level of accuracy is difficult to obtain with an atomic clock. It's unheard of for a field surveying device. A more precise method is needed.

Most EDM devices solve this problem by using the laser as the clock. The laser pulse has a frequency. That frequency corresponds to a wavelength. If we can measure the offset between the transmitted and received signal, we can measure the time delay.

The wavelength of a signal can be derived from the frequency using the formula:

$\lambda = c / h$

Where:

h = frequency

λ = wavelength

c = the speed of light

Given the wavelength, the distance can be calculated using the formula:

$P = (N \lambda + d) / 2$

Where:

N = the number of full wavelengths received at the detector.

d = the fractional part of a wavelength received.

Note that the higher the frequency of the signal, the greater precision in the distance measurement.

The fractional part of the wavelength is the phase shift (figure —-). While high-precision clocks are difficult to build, a tuning circuit capable of measuring the phase shift is simple and inexpensive.

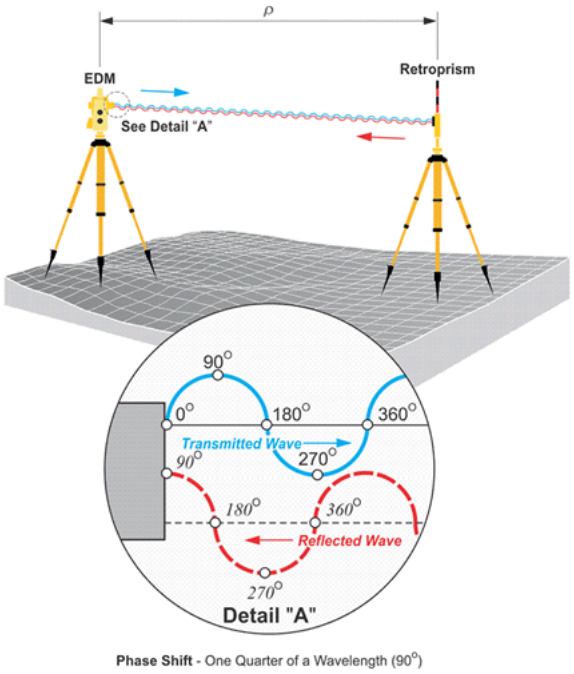

Figure 28 — Phase Shifting in EDM

But how do we solve for N? Counting cycles is impractical, particularly since many EDM devices use a continuous wave. The solution is to measure the distance using multiple frequencies. Since lower frequencies have a longer wavelength, we can start with a low-frequency, low-resolution measurement, then incrementally increase the frequency, thereby refining the measurement.

Another approach is to encode a pseudo-random sequence onto the signal. The sequence in the reflected signal is then compared to the original. Since we know when the signal was transmitted, any miss-alignment between the reflected sequence and the original indicates the elapsed time (Δt). If the sequence is long enough to span multiple cycles, then N can be found by multiplying Δt by the frequency (h) and rounding down to the nearest whole cycle:

$N = \Delta t * h$

Since the phase shift approach is more precise, most implementations use a code sequence to measure N and phase shift to measure d.

6.1.4. GPS

The Global Positioning System (GPS) is the most widely known precise positioning technology we have today. Yet, the GPS satellites obit 20,183 km above the earth surface. How can something so far away provide measurements so precise?

6.1.4.1. GPS Time

An understanding of precise positioning with GPS first requires an understanding of GPS time.

The GPS system consists of a constellation of Earth orbiting satellites. Each satellite is fitted with a highly accurate atomic clock, which is periodically synchronized by a ground control station located at USNO, Colorado. As a result, the GPS satellites share a single synchronized temporal reference system. This temporal reference system is GPS time. USNO ensures that GPS time has an accuracy of ≤40 nanoseconds 95% of the time.

The GPS time scale consists of two parts. The first part is a count of the number of weeks since the epoch. Each GPS week is 604,800 seconds long. Since GPS is a monotonic reference system, it does not include leap seconds or years. The second part is the number of seconds in the current week. The start epoch is 0 hours (midnight) Sunday 6-Jan-1980, when GPS time was 0.

While the atomic clocks used in GPS satellites are good, they are not perfect. They tend to drift off perfect alignment with GPS time. Furthermore, frequent resetting would degrade the lifespan of the clocks. So, GPS satellites also record the clock bias (τ), the difference between GPS and Space Vehicle (SV) time. This information is provided to the receiver in the NAV message.

There are a few rules governing the use of SV time:

1. Each SV operates on its own SV time,
2. All time-related data in the NAV messages shall be in SV time,
3. All other data in the NAV message shall be relative to GPS time,
4. The acts of transmitting the NAV messages shall be executed on SV time.

6.1.4.2. GPS Signal

GPS signals are driven by the on-board atomic clocks. Four frequency bands are used (see figure —)

Table 2

Band	Frequency
L1	1575.42 MHz
L2	1227.60 MHz
L3	1381.05 MHz

L5	1176.45 MHz

The L1 and L2 bands serve as carriers for broadcasting GPS data to GPS receivers. A carrier is not intended to convey information. It serves as a medium upon which other signals can be superimposed. This is the same principle as an FM radio. Your radio is tuned to a carrier frequency. The sound you hear is a separate signal which is superimposed or encoded onto the carrier signal. In the case of GPS, three additional signals are transmitted over the carrier:

Navigation Message: The Navigation Message (NAV) provides the receiver with metadata about the satellite. It is broadcast at 50 bps and takes about 30 seconds to transmit. This message includes the satellite ephemeris data, satellite clock corrections, almanac data, ionosphere and troposphere corrections, and satellite health data.

C/A Code: The C/A code is a 1023-bit pseudo-random number that repeats every 1 ms. The C/A code is broadcast at the rate of 1.023 Mbps. It has a chip length (distance between binary transitions) of 293 meters. Given a 1023-bit code and a chip length of 293 meters, the C/A sequence repeats every 300 km. (1023 * 293). Its primary purpose is to identify the satellite and to phase-lock the receiver and satellite clocks.

P Code: The P code is a pseudo-random number that repeats every 37 weeks. Each GPS satellite is assigned a one-week section of the P code. This section serves as a unique identifier, which helps a GPS receiver distinguish one satellite's transmission from another. Each satellite broadcasts its' section of the P code at the rate of 10.23 Mbps. It has a chip length (distance between binary transitions) of 29.3 meters and repeats every seven days. Due to the higher resolution possible at this higher broadcast rate, the P code may be encrypted.

In addition to the signals generated by the satellite, GPS receivers generate the same signals based on their own clock. These signals are used to correlate the signals received from the satellite with the local conditions at the receiver.

6.1.4.3. Pseudo Ranging

GPS positioning is based on trilateration. Trilateration calculates a location using three or more control points and the distances to each of those control points (figure —). In the case of GPS, the control points are satellites located at P_1, P_2, and P_3. The GPS receiver is located at A. If we construct a sphere around each control point (P_i) of radius (L_i), then the location of A is at the intersection of the three spheres.

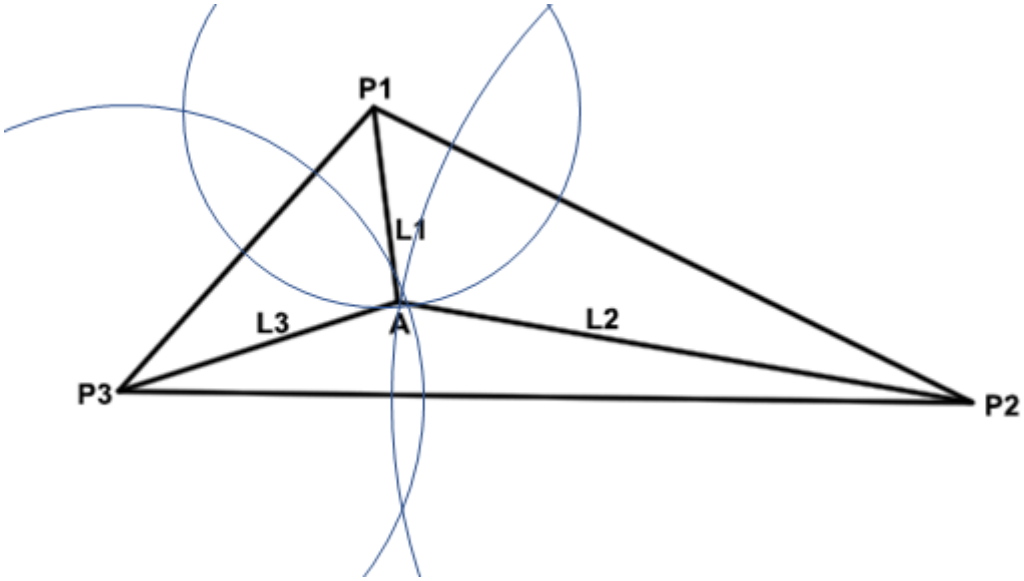

Figure 29 — Pseudo Ranging in GPS

Therefore, all we need to know for satellite-based precise positioning is the locations of P_i and the distances L_i. Much easier said than done.

GPS satellites are tracked to a high degree of precision by the GPS Control Segment. This "ephemeris" data is sent to the satellite every 4 hours. Receivers use a standard "Ephemeris Algorithm" to convert this data into an Earth-centered cartesian (x,y,z) coordinate in the WGS-84 coordinate reference system. However, since the satellites are moving, the calculated position is only valid for a specific time instance. A 1 nano-second (1.0×10^{-9} seconds) error in time can yield a 30 cm error in range.

GPS works on the same basic principles as an EDM device. Unlike an EDA, however, a GPS signal is one-way. The transmitted signal cannot be directly compared with the received signal. So the receiver first calculates the pesudorange observable and iterates to find an accurate solution.

The pseudorange is calculated by taking the time required for the signal to reach the receiver and multiplying that value by the speed of light. The basic formula to calculate a pseudorange is:

$P_i = c * (t_a - t_i)$

Where:

P = the pesudorange for satellite "i"

c = the speed of light

t_a = the time at position A when the signal was received.

t_i = the time on satellite "i" when the signal was transmitted.

The pseudorange can also be defined in terms of the locations of the satellite and receiver.

$P_i = ((x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2)^{1/2} + c(\tau) - c(\tau_i)$

The terms $((x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2)$ are an application of the Pythagorean Theorem ($a^2 + b^2 = c^2$). The terms $c(\tau)$ and $c(\tau_i)$ are the range error introduced by the clock bias at the receiver and on the satellite respectively.

P_i was calculated using equation —-. The satellite location and clock bias was provided through the NAV message. This leaves us with four unknowns. The location of A (x, y, z) and the receiver clock bias (τ). If we are working with four satellites, then that gives up four equations with four unknowns. These four simultaneous equations are usually solved using a least squares method. The result is a reasonably accurate position for A in x, y, z, and t. Repeating this process will refine the results. In particular, improved accuracy of the receiver clock will result in better range accuracy.

6.1.4.4. Carrier Phase Observable

The pseudo ranging result can be further improved using the Carrier Phase Observable. This approach is similar to the phase shift technique employed by EDM devices. The main difference is that the received signal is compared to the locally generated reference signal rather than the reflection. This gives us the formula:

$P_i = N_i \lambda + d_i$

Where:

N_i = the number of full wavelengths received at the detector.

λ = wavelength of the carrier

d_i = the fractional part of a wavelength received.

We can also represent the carrier phase observation using the Carrier Phase Bias (B_i) where:

$B_i = \lambda (\Phi - \Phi_i - N_i)$

Where:

Φ = the phase generated by the receiver clock

Φ_i = the phase of the incoming signal

Adding the Carrier Phase Bias to the pseudorange gives us:

$P_i = ((x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2)^{1/2} + c(\tau) - c(\tau_i) + B_i$

Once again we have four equations solving for four variables: x, y, z, and τ, but with the extra precision added by B_i.

6.2. 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 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 (days, weeks, years) 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 than 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.

6.2.1. ISO 19108

The ISO standard for Temporal Reference Systems is ISO 19108:2006 Geographic Schema — Temporal Schema.

In our discussion of Moving Features, we found that time-variant properties of Features were selected using a TM_Coordinate parameter. TM_Coordinate is defined in ISO 19108 and illustrated in Figure —-

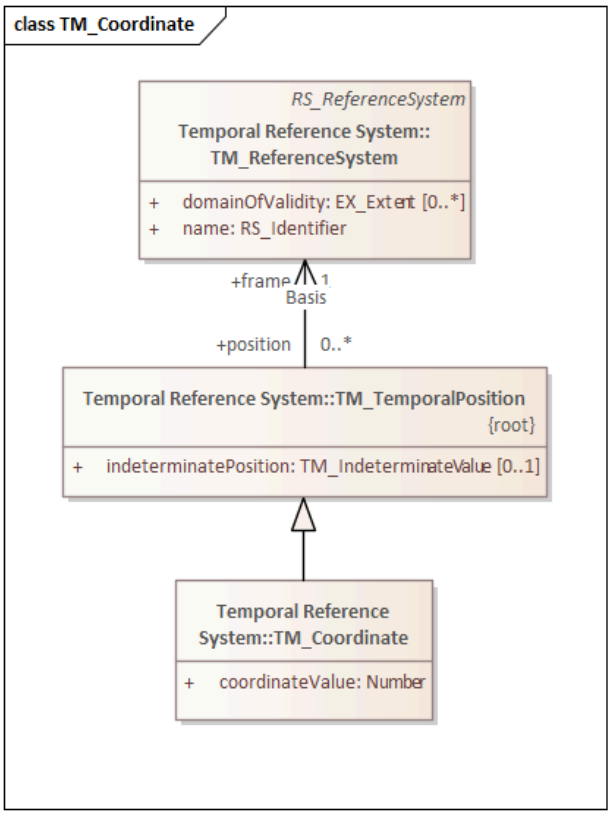

Figure 30 — TM_Coordinate UML

From this fiqure we see that a TM_Coordinate is a type of TM_TemporalPosition. And that each TM_TemporalPosition is associated with the TM_ReferenceSystem within which it is measured. So to understand Temporal Reference Systems as defined in ISO 19108, we must explore the capabilites and limits of TM_ReferenceSystem.

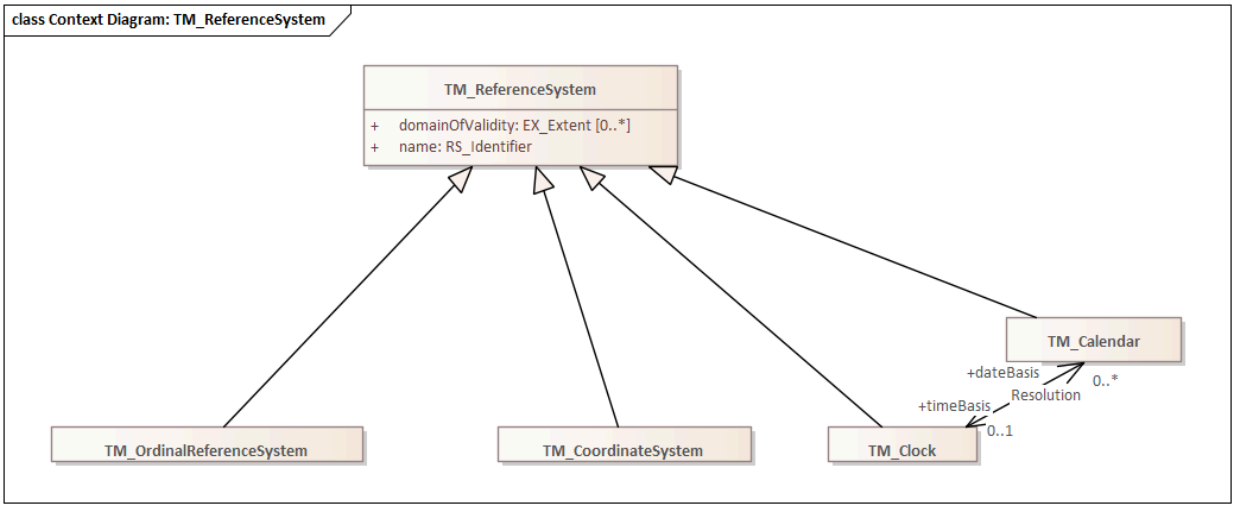

Figure 31 — TM_ReferenceSystem UML

We see from Figure — that there are four types of TM_ReferenceSystems. Since our concern is with time, not date, the TM_CoordinateSystem and TM_Clock variants are of most interest.

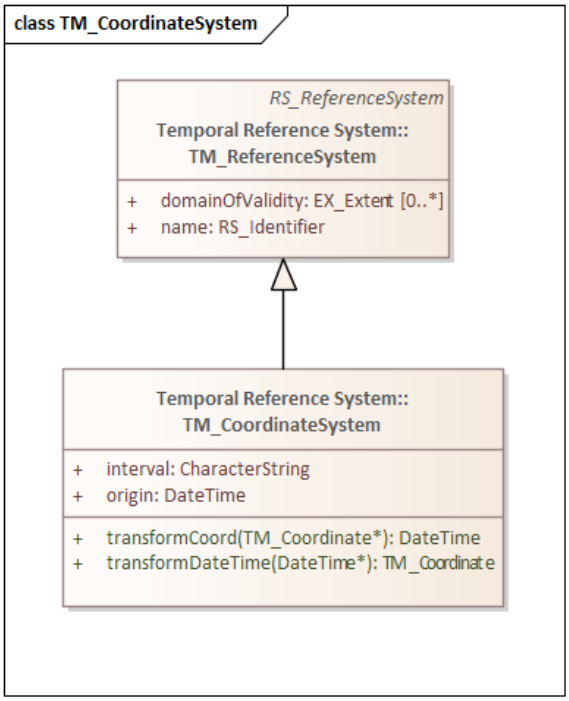

Figure 32 — TM_CoordinateSystem UML

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.

However, TM_CoordinateSystem does have limitations. The transformation operations, for example, only convert to and from DateTime. DateTime is a primitive type defined in ISO 19103. These operations cannot be used to convert from one TM_CoordinateSystem to another. Furthermore. TM_CoordinateSystem does not provide sufficient information to perform these transformations through an outside service.

The other alternative is TM_Clock illustrated in Figure —

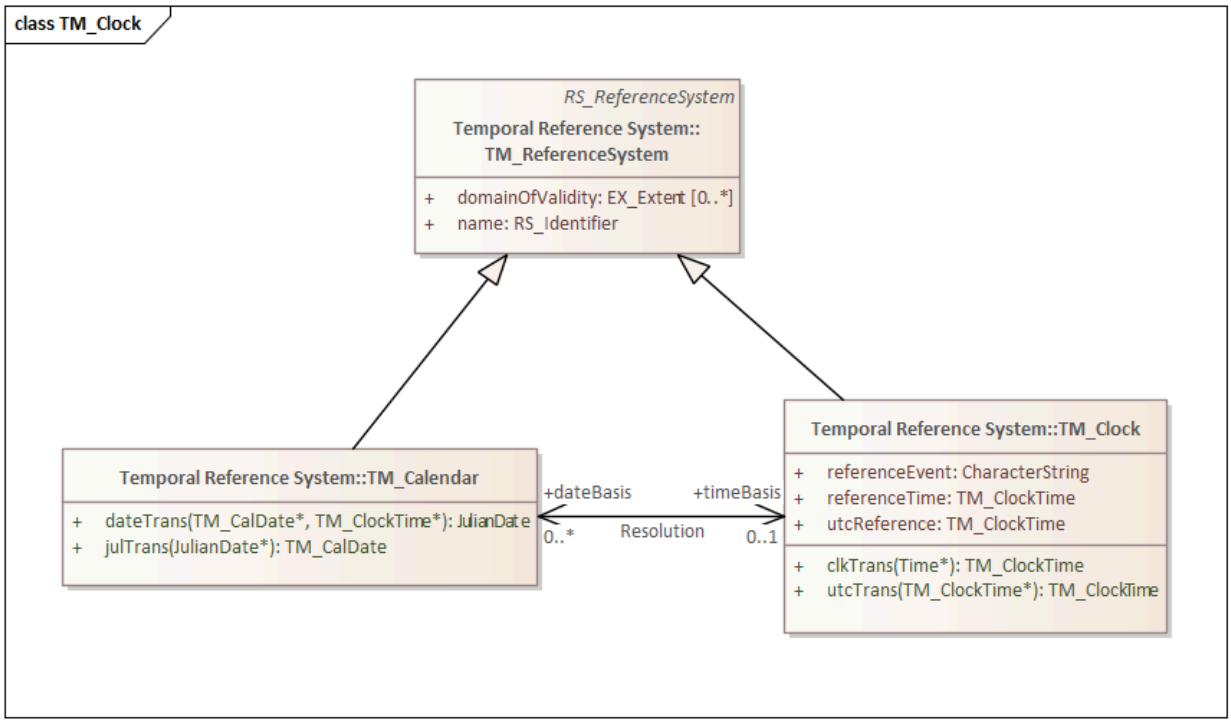

Figure 33 — TM_Clock UML

TM_Clock is "A system for measuring temporal position within a day". It has an optional `dateBasis` association with a calendar (TM_Calendar) class. This allows a TM_Clock to be associated with a TM_Calendar to create a full date-time reference system.

TM_Clock posesses the following attributes and operations:

- referenceEvent — The name or description of an event, such as solar noon or sunrise, which fixes the position of the base scale of the clock.
- referenceTime — The time of day associated with the referenceEvent, expressed as time on this TM_Clock.
- utcReference — The time of day associated with the referenceEvent expressed as a time in UTC or a related standard time.
- clkTrans() — This operation accepts a time expressed in UTC and returns a time expressed in this TM_Clock.
- utcTrans() — This operation accepts a time of day on TM_Clock and returns a time expressed as UTC or a related standard time.

The referenceEvent and referenceTime attributes are particularly interesting. Since these are not restricted in scope, they can be used to define a temporal reference system with an arbitrary origin. For example, an airborn Motion Imagery collection mission can set the referenceEvent value to "takeoff" and the referenceTime to the value of the internal clock when the wheels left the runway.

Furthermore, the attributes and operation parameters for TM_Clock are almost exclusively TM_ClockTime classes. A TM_ClockTime is the “time of day measured in a time keeping system other than UTC.” Its sole attribute, clkTime, is a sequence of numbers. A suitable representation for a local nanosecond clock.

This combination of classes allows us support high precision local-clock TRS as well as full date-time TRS. The only issue is that support for UTC time is required, even for non-terrestrial clocks.

At this point it appears that transformations between clocks is not supported by 19108. But there is another avenue we can explore. TM_CoordinateSystem is a subclass of TM_ReferenceSystem. And TM_ReferenceSystem is a subclass of RS_ReferenceSystem. This takes us into ISO 19111: Referencing By Coordinates.

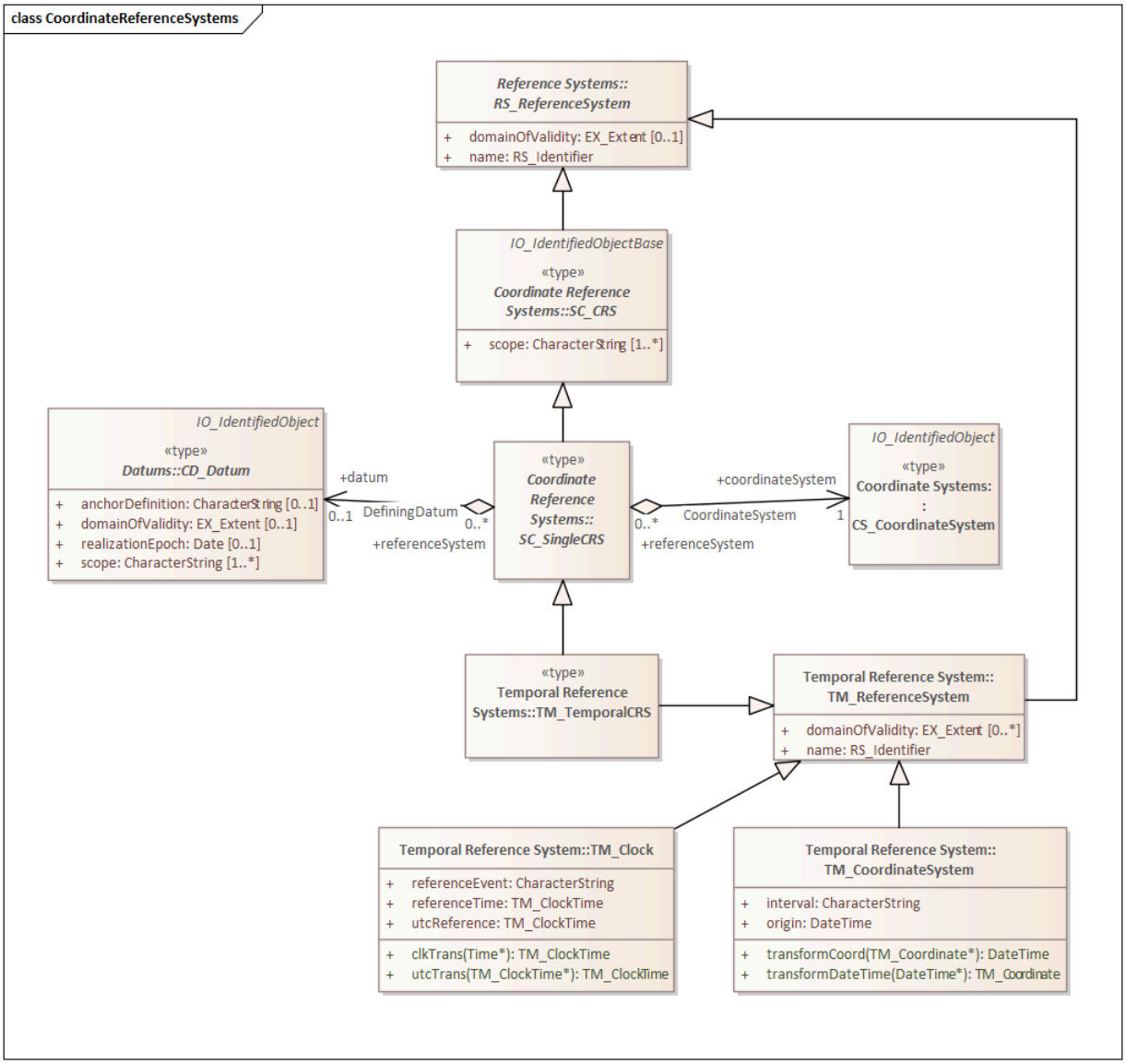

Figure 34 — Coordinate Reference Systems

In this figure, we see that a TM_ReferenceSystem is an RS_ReferenceSystem as defined in ISO 19111. But it is not an SC_CRS. As such, it does not inherit the additional properties defined in

the subclasses of SC_CRS. But we do have a way out. TM_TemporalCRS (defined in 19111 of all places) is a subclass of both SC_SingleCRS (19111) and TM_ReferenceSystem (19108). So if we want to promote Time to a first-class dimension, then TM_TemporalCRS is a good place to start.

The question is, what do we need from a temporal coordinate reference system?

6.2.2. 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 nanosecond counter.

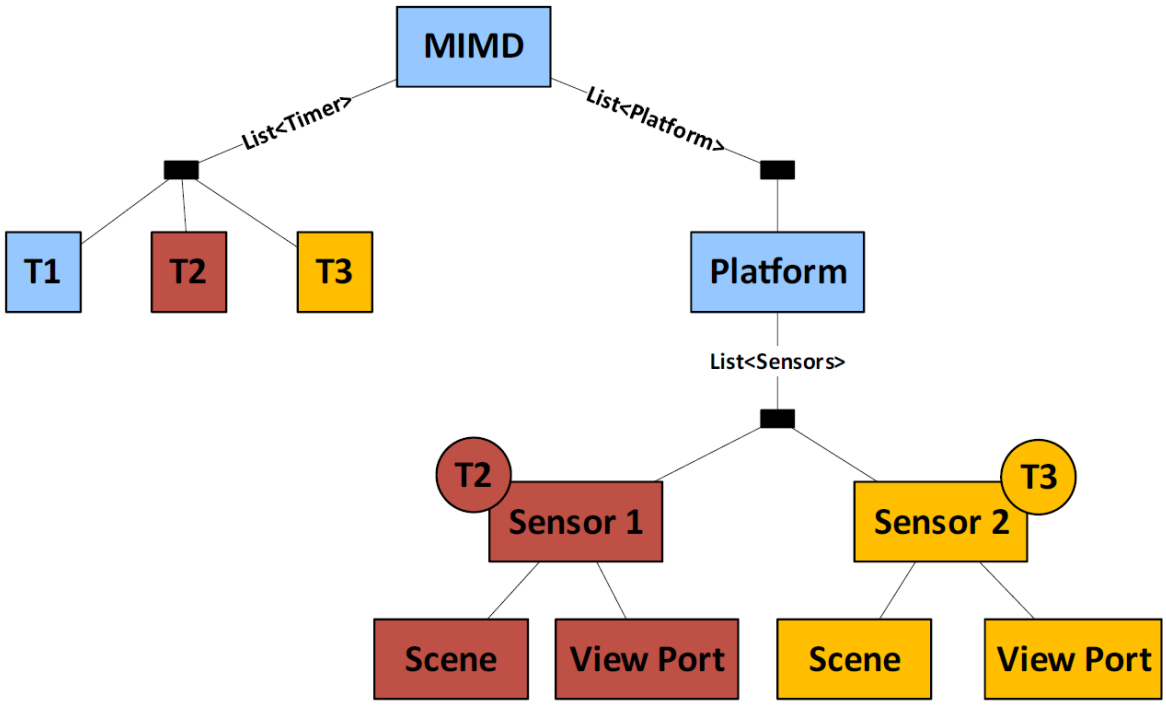

Figure 35 — MISB Timer Refence Systems

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.

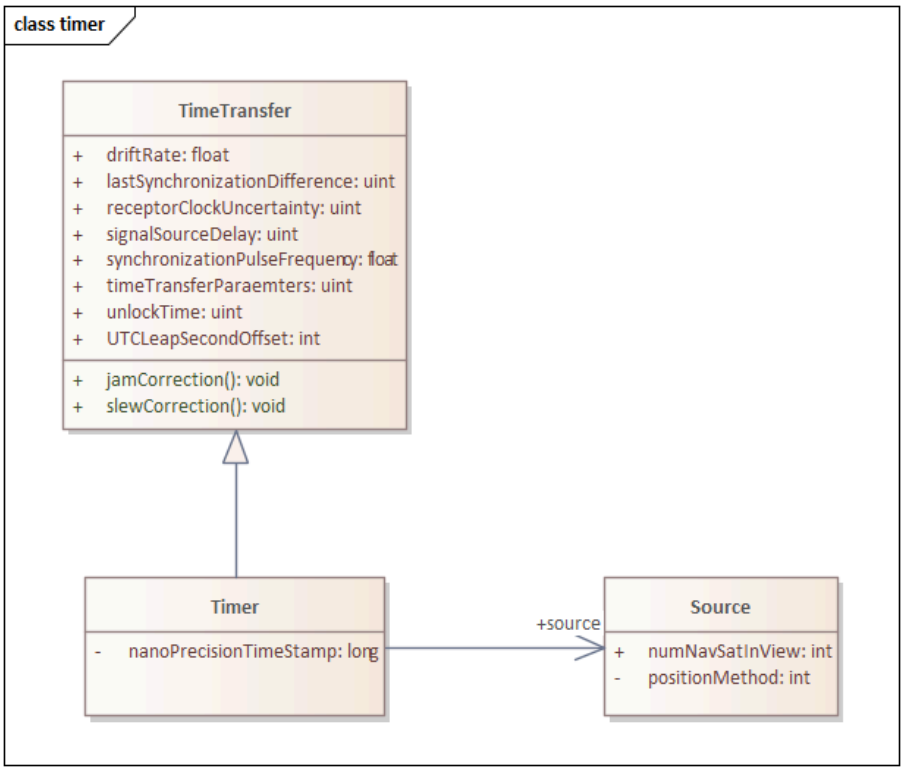

Figure 36 — MISB Timer UML Model

figure — illustrates the Timer architecture. A Timer is a physical device which provides a time stamp with up to nanosecond precision. The timer is associated with a “Source”. The Source specifies how the timer is initallized and how any subsequent corrections are applied. Finally, the TimeTransfer pack defines the metadata available to detect and apply any corrections to the time stamp.

The MISB timer concept is specific to the platforms and collection techniques used for Motion Imagery. These can be extended to provide a more general solution.

6.2.3. Discussion

ISO 19108 does not support tranformation between temporal reference systems. However, we can combined ISO 19108 with ISO 19111 to create a rudimentary transformation model for temporal reference systems. This approach will also allow us to minimize the difference between temporal and spatial reference systems.

One characteristic of temporal transformations is that the transformation parameters are not static. They are often based on the current state of a physical clock. This suggests that a temporal reference system may include a Moving Feature. A representation of a real-world object with Temporal Properties. The MISB Timer system is an example of this approach. However, the MISB approach is designed for the limited scope of motion imagery collection systems. It does not provide a general solution. Therefore, a more robust model for both Precision Temporal Reference Systems and their transformation is needed.

In summary:

1. We can define high-precision temporal CRS as subclasses of TM_TemporalCRS.
2. A high-precision temporal CRS is a physical device, a Timer.
3. Precision Time Stamps are created from the local Timer, which is a Temporal CRS
4. Since a Timer is a device, it is also a Feature.
5. Transformation of values from one timer to another requires knowledge of the distance between timers.
6. Since timers can move, they must be Moving Features
7. Transformation parameters may also be time sensitive. So Timers must support termporalProperties.
8. A photon travels roughly 30 centimeters in 1 nanosecond. To achieve and maintain 1 nanosecond accuracy, an accurate measurement of distance, and the associated adjustment, is required.
9. If the timer is traveling very fast, relativistic effects must be accomodated.

7

REFERENCE SYSTEMS

7 REFERENCE SYSTEMS

7.1. Planetary Reference Systems

NASA, USGA, and the IAU are developing Reference Systems for non-Earth planetary bodies.

ISO 19111 is the international standard for defining Reference Systems

Reference EPSG and NSG registry for examples

Work to do, further refinement of the existing planetary reference systems and develop ISO 19111 compliant descriptions of those reference systems.

Additional work to consider – The Space AOI is likely to require planetary Reference Systems consisting of a point mass with surrounding non-uniform magnetic and gravitational fields. Is this a valid requirement, how do we define such a reference system, and can be extend IAO 19111 to support it.

7.2. Astronomical Reference Systems

TBD — refer to the International Astronomical Union (IAU)

7.3. Local Reference Systems

TBD

7.4. Dynamic reference Systems

TBD

7.5. Space-Time Reference Systems

Minkowski Space Time is a four-dimension Reference System commonly used in Special and General Relativity. 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 ©

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.

7.5.1. 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 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.

8

SYSTEMS OF REFERENCE SYSTEMS

8 SYSTEMS OF REFERENCE SYSTEMS

8.1. Reference Systems Problem Statement

Traditionally we have taken reference systems for granted. Everything was ultimately referenced to the Earth. Not so in Space,

Mickelson Moorely shows us that there is no Universal Reference System. All measurements of location are relative. Object A measures the velocity of Object B relative to itself. This measurement is unique to the A-B relationship. There is no absolute measure of velocity that Object B can report to any obserer.

Reference System integration — Stages System, GeoPOSE, the need for more complex models

The need for a more robust concept of coordinate system transformation.

8.2. MISB Stages

8.3. GeoPOSE

8.4. Future and Ongoing Work

9

SPECIAL RELATIVITY

9 SPECIAL RELATIVITY

9.1. Dynamic Features at Relativistic Velocities

Newtonian physics, the physics of everyday life, served us well for almost 200 years. By the late 1800's, however, it became clear that something more was needed.

9.1.1. The birth of relativity

Since light behaves like a wave, there must be a material for the waves to propagate through. Physicists called this material the luminiferous aether. The aether was an undetectable substance which permiates space. It provided a universal reference frame against which absolute motion could be measured.

According to this theory, a measurement of the speed of light would vary depending on the motion of the measuring apparatus through the universal reference frame, the aether. The speed of light through the aether would be a fixed value. Measurements of the speed of light would be the sum of the velocity of the measurement appraratus relative to the aether and the speed of light through aether.

Consider a measuring apparatus which consists of a light separated a fixed distance from a detector. If the apparatus is oriented in the same direction as the motion of the apparatus through the aether, then the measured speed of light would be higher. If the apparatus was traveling in the opposit direction, then the measured speed of light would be lower.

should be higher for an object moving toward a light source and lower for one moving away.

In 1887, Michaelson and Morley built such an apparatus. Their experiments showed that the speed of light was the same no matter how their apparatus was oriented. The inescapable conclusion was that there is no luminiferous aether. That there is no universal reference system. There is no abolute motion. All motion must be a measurement of relative motion between two objects.

9.1.2. Lorentz Transformation

The Lorentz transformations were the result of attempts by Lorentz and others to explain how the speed of light could be independent of the underlying reference frame without discarding the luminiferous aether. While the aether theory is long discarded, Lorentz Transforms are consistent with Special Relativity which makes them still useful today.

Consider a reference frame 'A' which is moving in respect to an observer at 'B'. Lorentz asserts that 'A' and 'B' measure reality using two different reference frames. Transformation from reference frame 'A' (Fa) to Reference Frame 'B' (Fb) requires a transformation that accounts for

the effect of velocity. Central to this transformation is the The Lorentz factor (g) which is defined as:

g = 1 / (sqrt(1 - (v^2 / c^2)))

Figure 37

In each reference frame, an observer can use a local coordinate system (usually Cartesian coordinates in this context) to measure lengths, and a clock to measure time intervals. An event is something that happens at a point in space at an instant of time, or more formally a point in spacetime. The transformations connect the space and time coordinates of an event as measured by an observer in each frame.

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).

x' = g(x-vt) t' = g(t-(vx/c^2)) y' = y z' = z

Show how a velocity of 1/2 c will generate space and time contraction.

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.

9.1.3. 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 ©

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 $c^{2dt}2$ 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.

9.1.4. Musings

A Minkowski coordinate reference system is M4D.

Minkowski spacetime has no natural CRS. All M4D CRS are local.

M4D is a spatial CRS. the axis are x, y, z, and ct.

Note the units of ct → M/S * S = M → all axis of a M4D CRS have the same units of measure.

Consider an Observer O and an Observed phenomenon O'.

Discuss Worlds and world lines

If O' is stationary in respect to O, then Newtonian physics applies.

If O' is moving in respect to O, then it has a world line as shown in figure —-

Notice that the M4D CRS of O' is inclined in respect to the CRS of O. It has undergone a rotation. The effect of this rotation is a contraction in both the measured X and ct coordinates. The ammount of contraction depends on the rotation of M4D O' in respect to M4D O.

So M4D O' and M4D O can be evaluated in terms of our standard concepts of translation and rotation.

The actual mathematics involved can be packaged in a reusable library, similar to the way terrestrial CRS conversions are handled by GeoTRANS. It is sufficient for geospatial users to understand the required parameters and the basic principles of the transforms.

Also — this requires that the general practice of measuring across coordinate reference systems be matured. CRS and CRS transformation will play a much larger role in spatial-temporal data processing than it has in the past.

9.2. Discussions and Recommendations

TBD

Quaternions vs. Euler rotations

Temporal representation — Unix time vs the Timer model.

Standarize on a right-handed Cartisian coorinate system for all stages/Poses?

A

ANNEX A (INFORMATIVE) GLOSSARY

A ANNEX A (INFORMATIVE) GLOSSARY

The following terms and definitions are used in this Discussion Paper.

- **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.

- **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)
- **direct position:**
  DirectPosition object data types hold the coordinates for a position within some coordinate reference system.
- **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)
- **feature:**
  abstraction of real world phenomena (ISO 19101:2002, definition 4.11)

NOTE 3: A feature may occur as a type or an instance. Feature type or feature instance shall be used when only one is meant.

- **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)
- **geometric object:**
  spatial object representing a geometric set (ISO 19107:2003, definition 4.47)
- **geometric primitive:**

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

NOTE 4: Geometric primitives are non-decomposed objects that present information about geometric configuration. They include points, curves, surfaces, and solids.

- **global coordinate system:**
  the CRS of the trajectory of the reference point.

NOTE 5: 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]

- **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.
- **Inertial Frame:**
  relative motion with constant velocity
- **instant:**
  0-dimensional geometric primitive representing position in time (ISO 19108:2002, definition 4.1.17)
- **leaf:**
  a one parameter set of geometries. The geometry at a particular value of the parameter (ISO 19141:2008, definition 4.1.12)
- **local coordinate system:**
  a coordinate reference system which 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.
- **Non-Inertial Frame:**
  accelerating, moving in curved paths, rotational motion with constant angular velocity, etc.
- **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 9141:2008, definition 4.1.15)

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

- **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 6: 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.

- **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)
- **temporal position:**
  location relative to a temporal reference system (ISO 19108:2002, definition 4.1.34)
- **temporal reference system:**
  reference system against which time is measured (ISO 19108:2002, definition 4.1.35)
- **trajectory:**
  path of a moving point described by a one parameter set of points (ISO 19141:2008, definition 4.1.22)
- **vector:**
  quantity having direction as well as magnitude (ISO 19123:2005, definition 4.1.43)

B

ANNEX B (INFORMATIVE) REVISION HISTORY

B ANNEX B (INFORMATIVE) REVISION HISTORY

Table B.1

DATE	RELEASE	EDITOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
2016-04-28	0.1	G. Editor	all	initial version

BIBLIOGRAPHY

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)