OGC® DOCUMENT: 23-011R1

External identifier of this OGC® document: http://www.opengis.net/doc/PER/T18-D024

Open Geospatial Consortium

TESTBED-18: 3D+ DATA SPACE OBJECT ENGINEERING REPORT

ENGINEERING REPORT

PUBLISHED

Submission Date: 2023-02-02
Approval Date: 2023-03-03
Publication Date: 2023-06-26

Author: Chuck Heazel
Editor: Martin Desruisseaux, Logan Stark

Notice: This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is *not an official position* of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard.
Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

License Agreement

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 EXECUTIVE SUMMARY

This OGC Testbed 18 3D+ Data Space Object Engineering Report (ER) begins with the application of *ISO 19111: Geographic Information – Spatial referencing by coordinates* (ISO 19111:2019) to the reference frame of objects in space such as celestial bodies or spacecraft in orbit. The Data Space Object ER sits between the *3D+ Standards Framework Engineering Report* (OGC 22-036), which introduces the theoretical grounds, and the *Reference Frame Transformation Engineering Report* (OGC 22-038), which applies ISO 19111 to coordinate operations between above-cited frames. This ER explores the flexibility of the ISO 19111 model and concepts for adapting the coordinate system to the object geometry (cylindrical, triaxial ellipsoid, *etc.*).

Coordinate Reference Systems (CRSs) are sometimes inseparable from the coordinate operation that relates them to another system. This is the case of the ISO 19111 concept `DerivedCRS` class where by *definition* a CRS is specified in terms of its relationship with another CRS. This situation is also the case in the OGC GeoPose 1.0 Data Exchange standard (OGC 21-056r10) which describes a chain of inner frames related to outer frames by coordinate operations. Due to this interdependency, both derived CRS and GeoPose are discussed in OGC 22-038.

According to this ER, the ISO 19111 Standard has the richest set of capabilities for expressing complex chain of coordinate operations while preserving information about changes of datum. The latter are constraints of the physical world. Most concepts developed in GeoPose Standard and in NASA SPICE software can be mapped to ISO 19111 concepts. GeoPose provides a more compact encoding for the use cases of interest to GeoPose.

II KEYWORDS

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

ogcdoc, TB-18, CRS, 3D+, space-time

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):

- Blue Monocle, Inc.
- Geomatys

V CONTRIBUTORS

Editors and contributors to this Document include the following:

- Martin Desruisseaux (Geomatys) — editor
- Charles Heazel (Heazeltech) — contributor
- Logan Stark (Blue Monocle, Inc.) — editor

VI ABSTRACT

With the growing commercialization of space there is a need to look beyond the earth and explore the integration of sensors or assets in celestial orbits or in free flight in our solar system. Their exact tracking and localization are becoming increasingly important as space emerges as the newest area in need for standard-based mechanisms for streaming and for data integration from various sensors.

This Open Geospatial Consortium (OGC) Testbed 18 3D+ Data Space Object Engineering Report (ER) describes existing standards in terms of their ability to represent a suite of multidimensional Coordinate Reference Systems (CRS) and associated geometries as well as identifies shortfalls in these standards.

1

SCOPE

1 SCOPE

This OGC Testbed 18 Engineering Report provides information about the representation of coordinate reference systems (CRSs) using existing standards. The ER describes how the frame definition can be adapted to different kinds of object geometry such as cylinders. The ER does not cover the transformations between two or more reference frames. This topic is covered by the *Reference Frame Transformation Engineering Report* (OGC 22-038). The theoretical grounds for reference frames are covered in the OGC *3D+ Standards Framework Engineering Report* (OGC 22-036).

2

NORMATIVE REFERENCES

2 NORMATIVE REFERENCES

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO: ISO 19107:2003, *Geographic information — Spatial schema*. International Organization for Standardization, Geneva (2003). https://www.iso.org/standard/26012.html.

ISO: ISO 19109:2015, *Geographic information — Rules for application schema*. International Organization for Standardization, Geneva (2015). https://www.iso.org/standard/59193.html.

ISO: ISO 19111:2019, *Geographic information — Referencing by coordinates*. International Organization for Standardization, Geneva (2019). https://www.iso.org/standard/74039.html.

ISO: ISO 19136:2007, *Geographic information — Geography Markup Language (GML)*. International Organization for Standardization, Geneva (2007). https://www.iso.org/standard/32554.html.

ISO: ISO 19141:2008, *Geographic information — Schema for moving features*. International Organization for Standardization, Geneva (2008). https://www.iso.org/standard/41445.html.

ISO: ISO 19162:2019, *Geographic information — Well-known text representation of coordinate reference systems*. International Organization for Standardization, Geneva (2019). https://www.iso.org/standard/76496.html.

Carl Stephen Smyth: OGC 21-056r10, *OGC GeoPose 1.0 Data Exchange Draft Standard*. Open Geospatial Consortium (2022). https://docs.ogc.org/dis/21-056r10/21-056r10.html.

3

TERMS, DEFINITIONS AND ABBREVIATED TERMS

3 TERMS, DEFINITIONS AND ABBREVIATED TERMS

This document uses the terms defined in OGC Policy Directive 49, which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this document and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications (OGC 08-131r3), also known as the ‘ModSpec’. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

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

3.1. coordinate

one of a sequence of numbers designating the position of a point

Note 1 to entry: In a spatial coordinate reference system, the coordinate numbers are qualified by units.

[**SOURCE:** ISO 19111:2019]

3.2. coordinate reference system

coordinate system that is related to an object by a datum

Note 1 to entry: Geodetic and vertical datums are referred to as reference frames.

Note 2 to entry: For geodetic and vertical reference frames, the object will be the Earth. In planetary applications, geodetic and vertical reference frames may be applied to other celestial bodies.

[**SOURCE:** ISO 19111:2019]

3.3. coordinate tuple

tuple composed of coordinates

Note 1 to entry: The number of coordinates in the coordinate tuple equals the dimension of the coordinate system. The order of coordinates in the coordinate tuple is identical to the order of the axes of the coordinate system.

[**SOURCE:** ISO 19111:2019]

3.4. earth-centered inertial

a reference frames having its origin at the center of mass of Earth and which is fixed with respect to the stars

Note 1 to entry: For objects in space, the equations of motion that describe orbital motion are simpler in a non-rotating frame such as ECI.

3.5. earth-fixed

a reference frame which remains fixed with respect to Earth’s surface in its rotation, and then rotates with respect to stars

Note 1 to entry: Geodetic CRS defined by ISO 19111 are Earth-Fixed.

3.6. engineering coordinate reference system

coordinate reference system based on an engineering datum

Note 1 to entry: For example, a coordinate reference system local to a moving object such as a ship or an orbiting spacecraft.

[**SOURCE:** ISO 19111:2019]

3.7. engineering datum

datum describing the relationship of a coordinate system to a local reference

Note 1 to entry: Engineering datum excludes both geodetic and vertical reference frames.

[**SOURCE:** ISO 19111:2019]

3.8. reference frame

parameter or set of parameters that realize the position of the origin, the scale, and the orientation of a coordinate system

[**SOURCE:** ISO 19111:2019]

3.9. Minkowski spacetime

a combination of three-dimensional Euclidean space and time into a four-dimensional manifold where the spacetime interval between any two events is independent of the inertial frame of reference in which they are recorded

3.10. Abbreviated terms

CRS	Coordinate Reference System
CS	Coordinate System
CSV	Comma Separated Values
ECEF	Earth-Centered — Earth-Fixed
ECI	Earth-Centered Inertial
GML	Geographic Markup Language
ICRF	International Celestial Reference Frame
WKT	Well-Known Text

1D	One-dimensional
3D	Three-dimensional
4D	Four-dimensional

4

INTRODUCTION

4 INTRODUCTION

Currently, most OGC Standards focus on data that is observed on the surface or directly above planet Earth. Other standards, such as GeoSciML, look into the planet and provide a data model and transfer standard for geological data. There has been less focus on extra-terrestrial space and the exact location of remote sensors. The goal of this Engineering Report (ER) is to evaluate the current standards and standard-based mechanisms used to describe location, trajectory, and orientation of objects in orbit or free flight in our solar system. A separate OGC Testbed 18 ER explores standard-based mechanisms to transform a location within a reference frame to a location within another reference frame (OGC 22-038). Another point of interest is to stream and integrate data from various sensors mounted on devices that are described in different reference frames and located anywhere in the solar system (OGC 22-035).

Data structures for reference frames and coordinate systems are defined by the ISO 19111:2019 – *Referencing by Coordinates* Standard (ISO 19111:2019). This joint OGC/ISO Standard is also published as an OGC Abstract Specification, Topic 2: referencing by coordinates. In most sections of the Data Space Object ER, descriptions of concepts are accompanied by descriptions of their relationships with ISO 19111. ISO 19111 terms will be shown in bold characters when first defined.

5

SPATIAL CRS

5 SPATIAL CRS

An ISO 19111 **Coordinate Reference System** (CRS) is a combination of a reference frame (Clause 5.1) with a coordinate system (Clause 5.2), possibly completed by conversion parameters (typically map projections). The reference frame identifies the location and motion of CRS origin, while the coordinate system provides information about axes such as their units of measurement.

The relationship between CRS *types* and Reference Frame *types* is close to (but not exactly) a one-to-one relationship. For example, a `GeodeticCRS` is always associated with a `GeodeticReferenceFrame`. However, the relationship between CRS *instances* and Reference Frame *instances* is not a one-to-one relationship. There are possibly hundreds of CRS instances associated with the same Reference Frame instance. For example, there are hundreds of `ProjectedCRS` instances associated with the same "WGS 84" reference frame instance in the EPSG geodetic dataset. Those CRS instances differ by the map projection parameters, axis directions, or units of measurement.

The relationship between CRS types and Coordinate System types is more flexible. For example, a `GeodeticCRS` can be associated with an `EllipsoidalCS`, `SphericalCS`, or `CartesianCS`. This flexibility is discussed in Clause 5.2.

Thousands of reference systems have been specified for use around the world or in specific regions and for various purposes. The multitude of CRSs has led to the creation of the EPSG registry (EPSG) which provides unique codes for objects such as CRSs, ellipsoids, and datums. The EPSG registry also provides information about how to transform coordinates from one CRS to another. Those transformations are discussed in the Reference Frame Transformation OGC 22-038 ER.

5.1. Reference frames

A reference frame represents the position of the origin, the scale, and the orientation of a coordinate system. However that information is not encoded directly in the reference frame. ISO 19111 reference frames are identified only by their names, sometimes accompanied by information such as ellipsoid axis lengths or temporal epoch. The information contained in ISO 19111 reference frames is usually not sufficient for transforming coordinates between different frames without the help of an external database such as EPSG. This limitation is by design, as it would be very difficult to achieve reference frame descriptions comprehensive enough for inferring accurate transformations in all cases. In the past, attempts were made, notably with a `TOWGS84` data structure associated with each frame description. That data structure was a WKT element in Well-Known Text (WKT) 1 or a `CS_WGS84ConversionInfo` class in OGC 01-009 with 7 parameters: the rotation terms, translation terms, and a scale factor to apply on geocentric coordinates. The use of WGS84 as a hub for inferring arbitrary transformations is called *early-binding*. That approach is in contrast with the requirement to use an external database. This approach is called *late-binding* because no parameters are available until *both* source and target reference frames are known. These two approaches are not equivalent. There are Earth-based examples where the early-binding approach has unavoidable errors of a few meters while the late-binding approach is accurate to a few centimeters (Annex A). Consequently, it is not sufficient to define reference frames as standalone objects. An organization must also define a registry equivalent to EPSG with reference frame definitions together with *late-binding* information.

5.1.1. Reference frame types

Spatial reference frames can be classified broadly into the three following classes (OGC 22-036).

- *Space-fixed.* This class includes the Earth-Centered Inertial (ECI) frames.
- *Planet-fixed.* This class includes the Earth-Centered — Earth-Fixed (ECEF) frames.
- *Local system.* This class includes frames attached to spacecrafts.

An inertial frame is non-rotating with respect to the stars. Its velocity may be non-zero. For example the Earth is orbiting around the Sun, but the frame acceleration is considered negligible. Truly inertial frames are difficult to achieve in practice, so the “inertial” word in this ER should be understood as “quasi-inertial”.

For objects in space, the equations of motion that describe orbital motions are simpler in inertial (space-fixed) frames such as Earth-centered inertial (ECI). By contrast, to represent the positions and velocities of terrestrial objects, it is more convenient to use Earth-centered — Earth Fixed (ECEF) coordinates such as latitude, longitude, and altitude.

Each of above-cited classes can be further subdivided. For example, both spatial-fixed and planet-fixed frames have *“conventional (inertial or terrestrial) reference system”* and *“true (inertial or terrestrial) reference system”* sub-classes. Those sub-divisions reflect which effects are taken into account, for example precession or nutation.

This hierarchy of reference frames is detailed in the Testbed 18 OGC 22-036 ER. This hierarchy is not reflected in the ISO 19111:2019 standard, except for a separation between planet-fixed frames (**geodetic reference frame**) and local systems (**engineering datum**).

5.1.2. Local systems

A local reference system can be used for a local area on a planet surface or for a moving object such as a spacecraft. The OGC 22-036 ER subdivides local systems into different classes reflecting different choices of north or zenith directions. In ISO 19111:2019 the distinction is rather based on whether the local system is defined by a conversion from another, usually geodetic, coordinate reference system (CRS). This is often the case for a platform fixed on Earth, in which case the local system is a **derived CRS**. Conversely if the position of the local system is not defined in relationship with another system, for example if the local system is attached to an object in free flight, this is an **engineering CRS**. Note that the use of an engineering CRS does not mean that there is no relationship with another reference system, it only means that such relationship is not part of the CRS definition. The relationship may be empirical and based on observations having uncertainties.

5.2. Coordinate systems

The mathematics that can be applied for computing geometric properties in a given space depends on the type of **coordinate system** (CS). Most standards or toolkits examined by OGC 22-036, in particular NASA SPICE (SPICE) and the OGC GeoPose Standard, assume a **Cartesian CS**. ISO 19111 is more flexible as it allows choosing a coordinate system that is more convenient for the object geometry. For example, a rotating wheel space station (a.k.a. von Braun wheel) could use a **cylindrical CS**.

Some coordinate system types specified in ISO 19111 are listed below. This list is non-exhaustive, as a few additional coordinate system types exist for specific kinds of reference frame. For example, ISO 19111 also defines `TemporalCS` and `OrdinalCS` types, which are not considered in this section.

- `CartesianCS`
- `AffineCS` (Clause 5.2.4)
- `LinearCS` (ISO 19148:2021)
- `PolarCS`
- `CylindricalCS`
- `SphericalCS` (Clause 5.2.2)
- `EllipsoidalCS` (Clause 5.2.1)

The Coordinate Reference System (CRS) attached to a space station should be an `EngineeringCRS`, but the coordinate system associated to that engineering CRS can be any one from the above list except `EllipsoidalCS`. Below is an example in Well-Known Text (WKT) format applied to a rotating wheel space station.

```
ENGINEERINGCRS["Von Braun wheel",
  ENGINEERINGDATUM["Wheel rotation axis"],
  CS[cylindrical, 3],
  AXIS["distance (r)", awayFrom, ORDER[1], LENGTHUNIT["metre", 1]],
  AXIS["bearing (U)", clockwise, BEARING[234], ORDER[2], ANGLEUNIT["degree",0.0
174532925199433]],
  AXIS["height (h)", up, ORDER[3], LENGTHUNIT["metre", 1]]]
```

Figure 1 — Engineering CRS for a rotating wheel space station (WKT)

5.2.1. Ellipsoidal CS special case

Ellipsoidal coordinate systems are special in that applying mathematics with these systems requires knowledge of the semi-major and semi-minor axis lengths and that information is not part of the `EllipsoidalCS` class. Rather, that information is part of the `GeodeticReferenceFrame` class, through its `ellipsoid` association. Consequently, `EllipsoidalCS` can hardly be used as a standalone object. Its use requires the `GeodeticCRS` object which contains the `EllipsoidalCS` object. This particular relationship implies that attempts to use `EllipsoidalCS` with `EngineeringCRS` would provide incomplete information, because an `EngineeringCRS` can only be associated to an `EngineeringDatum` and the latter does not contain the `ellipsoid` association.

Whether the above restriction is a problem or not depends on whether there is a need to associate `EllipsoidalCS` with an `EngineeringCRS`. In other words, whether spacecrafts or other non-geodetic objects with ellipsoidal shapes are envisioned. If there is this need, a possible change to ISO 19111 could be to move the `ellipsoid` association from `GeodeticReferenceFrame` to `EllipsoidalCS`. However, the reason why the ellipsoid is currently associated with `GeodeticReferenceFrame` is because it was considered useful to keep information about the ellipsoid in, for example, a geocentric CRS using a spherical or Cartesian coordinate system. That information would be lost if the ellipsoid were associated with `EllipsoidalCS`.

5.2.2. Two-dimensional spherical CS

A spherical coordinate system is usually three-dimensional. However, the planetary domain sometimes needs to express spherical coordinates on a planet surface using only geocentric latitudes and longitudes, without a radius. If the surface is a sphere, then classical two-dimensional geographic coordinates can be used. However, if the surface is an ellipsoid, then the geocentric, geodetic, and astronomical latitudes differ (see OGC 22-036 figure 23). If the desired latitude is really geocentric rather than geodetic, then the coordinate system needs to be a `SphericalCS` rather than an `EllipsoidalCS`. In the current version of the ISO 19111 Standard, spherical coordinate systems must be three-dimensional. Therefore, a corrigendum is proposed which would support the definition of two-dimensional spherical coordinate systems. In the latter case, by convention, the point would be located on the planet surface. That surface corresponds to different radius values depending on the latitude. Computing those values requires information about the ellipsoid axes, which is another argument for keeping the `ellipsoid` association in `GeodeticReferenceFrame` (Clause 5.2.1) despite `EllipsoidalCS` being an apparently more natural location.

Note that the third-dimension (radius or ellipsoidal height) is relative to the planet center in a `SphericalCS`, while the third dimension is relative to the planet surface in an `EllipsoidalCS`. Consequently converting a two-dimensional coordinate tuple to a three-dimensional tuple on the planet surface can be done easily with `EllipsoidalCS`. The third coordinate is simply zero, regardless of the ellipsoid. But in the `SphericalCS` case, that conversion cannot be done without knowledge about the `Ellipsoid`.

5.2.3. Triaxial ellipsoid

ISO 19111 supports definition of triaxial ellipsoids with a “semi-median axis” attribute in addition to the “semi-major axis” and “semi-minor axis” (or “inverse flattening”) attributes. Triaxial ellipsoids can be associated to the `GeodeticReferenceFrame` in the same way as biaxial ellipsoids. However, this is a recent feature added in ISO 19111:2019 and not many libraries support triaxial ellipsoids yet.

5.2.4. Axis angles in affine CS

AffineCS is a generalization of CartesianCS where axes are not necessarily perpendicular. Like EllipsoidalCS, application of mathematics in AffineCS space requires additional information — the angles between pair of axes. The current ISO 19111 Standard does not provide this information directly. Encoding that information in axis directions is possible, but that solution would not be interoperable. For example, before the introduction of the BEARING keyword in WKT, axes of polar projections in the EPSG database were used to express directions such as *“South along 90 deg East”* instead of “East” or “North”.

Whether angle information is useful enough to be worth an addition in ISO 19111 is not yet determined. If this information is considered useful, a possible data structure could be an axisAngle attribute in the AffineCS class. The data type would be Angle and the occurrence would be 1 for a two-dimensional affine CS, or 3 for a three-dimensional affine CS. In the latter case the three angle values could be as below, in order:

- angle from first axis to second axis;
- angle from second axis to third axis; and
- angle from third axis to the first axis.

Implementation experiments would be required for checking if above proposal can work.

5.2.5. Cartesian CS right-handiness

While the CartesianCS class does not need to specify axis angles, it is still useful to know if the coordinate system is *left-handed* or *right-handed*. For example, the NASA SPICE software requires Cartesian coordinate systems to be right-handed. There is currently no easy way to get this information into the ISO 19111 model. Software could infer this information from axis directions but this can be difficult if the coordinate system uses uncommon axis directions such as “away from” or if directions such as “up” are non-obvious (as it may be in space). If the axisAngle attribute mentioned in Clause 5.2.4 is added, its values would be fixed to ±90° but the value signs may give indication about right-handiness. Alternatively, an easier extension could be to add to the CartesianCS class an isRightHanded attribute of type Boolean.

5.3. New spatial CRS types

The ISO 19111 model supports two of the three broad classes mentioned in Clause 5.1.1. They are in the two last rows of the following table. The class which is not yet supported is the space-fixed system. Adding this support to ISO 19111 model may be relatively easy by adding the inertial classes shown in the first table row.

Table 1 — Types of reference frames

SYSTEM CLASS	ISO 19111 REFERENCE FRAME CLASS	ISO 19111 CRS CLASS
Space-fixed	`InertialReferenceFrame` (proposal)	`InertialCRS` (proposal)
Earth-fixed	`GeodeticReferenceFrame`	`GeodeticCRS`
Local system	`EngineeringDatum`	`EngineeringRS`

The new inertial classes would be similar to their geodetic counterparts. This is like the `GeodeticCRS` class being associated with `CartesianCS`, `SphericalCS`, and `EllipsoidalCS`. As such, `InertialCRS` should allow associations to at least `CartesianCS` and `SphericalCS`. Whether `InertialCRS` should also allow association to `EllipsoidalCS` is yet to be determined.

Figure 18 in the OGC 22-036 ER shows sub-divisions of the above broad classes: conventional, mean, true, *etc*. There is currently no construct in the ISO 19111 model for this level of sub-division detail. While useful for the understanding of a chain of coordinate transformations, they may not be necessary for their execution by a software application. Nevertheless, if there is a need to represent this classification within the ISO 19111 model, it could be done, for example, with some naming convention applied on the value of the `name` attribute of Reference Frame classes. Alternatively, if the information is considered important enough, a code list could be added.

6

SPATIOTEMPORAL CRS

6 SPATIOTEMPORAL CRS

This chapter describes the use of ISO 19111 for combining time and space into one data point. Two different approaches are proposed, depending on whether the Newtonian physics is sufficient or if Einstein's relativity must be taken in account.

6.1. Galilean transformations

Newtonian physics, the physics of everyday life, served humankind well for almost 200 years. Galilean transformations, also called Newtonian transformations, relate the space and time coordinates (x, y, z, t) of two systems moving at a constant velocity relative to each other. Consider a reference frame F which is moving at velocity v_x in respect to an observer at F'. Assuming that the axes of both reference frames are parallel and that the velocity is along x and x' axes, then transformations from reference frame F to reference frame F' are as below:

- $t' = t$
- $x' = x - v_x \cdot t$
- $y' = y$
- $z' = z$

The fact that temporal coordinates (t) are independent of spatial coordinates (x, y, z) makes it possible to handle spatial and temporal components as two independent coordinate reference systems. This is the idea underlying the use of `CompoundCRS` in ISO 19111 for adding a temporal component to a spatial CRS. This separation is adequate for describing phenomena at speeds much smaller than the speed of light. Galilean transformations express the ideas that space and time are absolute, with measurements that are independent of the relative motion of the observer. This also implies that all speeds are relative, including the speed of light, which would depend on the observer motion.

6.2. Lorentz transformation

The 1887 ether-wind experiment of Michelson and Morley demonstrated that the speed of light is independent of the reference frame. Lorentz transformations are used for transforming coordinates between reference frames while preserving the light speed invariance. Consider a reference frame F which is moving in respect to an observer at F'. Transformation from reference frame F to reference frame F' requires a transformation that accounts for the effect of velocity. Central to this transformation is the Lorentz factor (γ) defined as:

$$\gamma(v) = 1 \Big/ \sqrt{1 - (v/c)^2} \qquad (1)$$

In each reference frame, an observer can use a local Cartesian coordinate system 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*, the coordinates are represented by (*x*, *y*, *z*, *t*). In the other reference frame *F* ′ moving at a velocity v_x relative to *F*, the coordinates are represented by (*x*′, *y*′, *z*′, *t*′). In both reference frames the coordinate axes are parallel, and they remain mutually perpendicular. The relative motion v_x is along the *x* and *x*′ axes and this section assumes that there is no motion along other axes. At *t* = *t*′ = 0, the origins in both reference frames are the same (*x*, *y*, *z*) = (*x*′, *y*′, *z* ′) = (0, 0, 0). At any other spatiotemporal coordinates, the transformations are as below:

- $t' = \gamma \cdot (t - v_x \cdot x / c^2)$
- $x' = \gamma \cdot (x - v_x \cdot t)$
- $y' = y$
- $z' = z$

The above equations imply length contraction and time dilation. In most cases those effects are negligible because γ is almost 1 and v^2/c^2 approaches 0. However, when *v* is a significant percentage of *c*, those equations should be applied.

The Lorentz contraction effect is the shortening of an object along the direction of its motion relative to an observer. Dimensions in other directions are not contracted. For applying above set of equations in practice, the *F*′ reference frame must be rotated and translated so as to be co-linear with *F*.

6.2.1. Length property in Moving Feature instances

Features (ISO 19109:2015) can contain an arbitrary number of properties. Some properties may be measurements of the length of something. Because of length contraction effects, the definition of linear properties in Moving Features must clarify how lengths are measured. The following are two possibilities.

- Given that the motion of a moving feature is described relative to some external CRS, the values of a linear property could be *lengths* as measured by an observer in that external CRS.
- Alternatively, moving features could express lengths as *proper lengths*. A proper length is the length as observed by an observer moving with the moving feature. This is equivalent to defining an `EngineeringCRS` for each Moving Feature instances and declaring that length measurements are relative to that CRS.

The second approach may seem to provide more natural length measurement, but may imply a greater amount of length transformations when a calculation compares the properties of different feature instances. The problem is somewhat similar to map projections. For example, if the trajectories of Moving Features on Earth are expressed in the Mercator projection, there is a possible ambiguity about whether the values of length properties in Moving Feature instances are real lengths or lengths as observed on a map using the Mercator projection. The latter measurements, despite being non-real, may be desirable for some types of comparison. For example, if the length property is a safety distance, then that length needs to be in the units of the map projection to detect when two Moving Features are closer than that distance (ignoring alternative approaches such as computing geodetic distances). On-the-fly transformations from real lengths to “Mercator” lengths are possible but may be costly for large number of Moving Features.

The case of Lorentz contractions is a bit different in that map projections are clearly mathematical artifacts, while Lorentz contractions have physical grounds. Historically, Lorentz contractions were an attempt to reduce the speed of light constancy to the status of an experimental artifact. However, Einstein’s relativity theory reversed that view by proposing that the speed of light is a universal constant. The similarity between Lorentz contractions and map projections is that a choice must be made between two alternatives (common CRS for all or implicit CRS for each feature instance), and the best choice depends on the problem to solve.

6.3. Minkowski space-time

Minkowski space-time 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*) multiplied by the speed of light (*c*). The concept equivalent to a distance in this coordinate system is the differential arc length, computed by $\Delta s^2 = \Delta x^2 + \Delta y^2 + \Delta z^2 - c^2 \cdot \Delta t^2$ where:

- Δx = change in *x* direction
- Δy = change in *y* direction
- Δz = change in *z* direction
- Δt = change in time

While a Lorentz transformation deals with spatial measurements, Minkowski space includes time as part of that spacetime. Thus, Δs is an arc length through spacetime as opposed to a difference in *x* as in the Lorentz transform.

Since $c^2 \cdot \Delta t^2$ is a negative term, it implies that *ct* can be handled as an imaginary number orthogonal to *x*, *y*, and *z*. That number can be *cti* such that $i^2 = -1$. Consequently an alternative form to above arc length equation is $\Delta s^2 = \Delta x^2 + \Delta y^2 + \Delta z^2 + (i \cdot c \cdot \Delta t)^2$.

The sum requires that all terms in Δs have the same unit of measurement. The *ct* term has a linear unit since (L/T)×T = L, but the units of *c* and *t* must be chosen in such a way that the product has the same unit as *x*, *y* and *z*.

Four dimensional vectors (Δx, Δy, Δz, $c \cdot \Delta t$) can be classified according to the sign of Δs. A vector can be *timelike*, *spacelike*, or *lightlike*. Timelike vectors are of special interest as they relate events that can have a causality relationship. The classification of any vector will be the same in all reference frames that are related by a Lorentz transformation.

Minkowski space-time support could be added to ISO 19111 with the addition of a new `MinkowskiCS` class. This proposal is discussed more extensively in OGC 22-038. The actual mathematics involved can be packaged in a reusable library. It is sufficient for geospatial users to understand the required parameters and the basic principles of the transforms. However, this requires that the general practice of measuring across coordinate reference systems be matured (Clause 6.2.1). CRSs and CRS transformations may play a larger role in spatiotemporal data processing than it has in the past.

6.3.1. Illustration

Consider Alpha Centauri. The distance to the Alpha Centauri galaxy is 4.37 light years. A light year is the distance a photon will travel through the vacuum of space in one year. This quantity is defined as the speed of light (*c*) multiplied by the travel time (*t*). The units of measurement are (m/s)×s = m. If the temporal axis is defined as *c* × *t* (*ct*), then (*x*, *y*, *z*, *ct*) can be defined as a single coordinate reference system. This is Minkowski space-time. Minkowski space-time is typically illustrated using a Minkowski diagram.

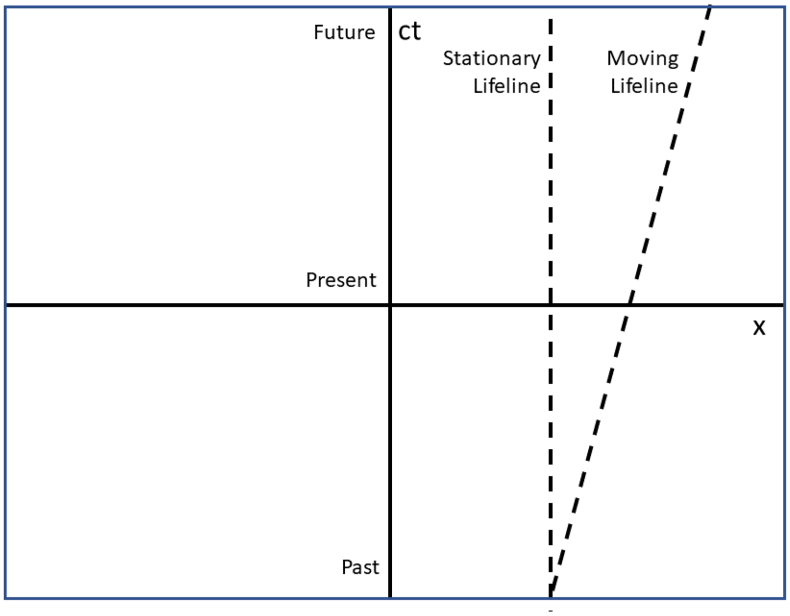

Figure 2 — Minkowski diagram

These diagrams contain two axes, the *x* (spatial) axis and the *ct* (temporal) axis. The temporal axis runs from the past (bottom) to the future (top). The *x* axis represents one of the spatial dimensions. Concepts illustrated using only the *x* axis are easily extended into *y* and *z* as well. Events in spacetime are captured as points on this graph.

Two lifelines are also included in this illustration. One is for an object which is stationary. The events that make up its lifeline accumulate along the *ct* axis but are stationary in the *x* axis. The other lifeline is for an object that is moving at a constant velocity. The events making up its lifeline accumulate in both the *ct* and *x* axes. Velocity is a measure of the slope of this lifeline.

6.4. Geometry and moving features

ISO 19141:2008 — *Schema for moving features* provides a framework for describing features having positions that change with time. While the standard name contains “moving features” words, the standard content is actually about moving geometries. The difference between a feature and a geometry is that the former contains arbitrary properties, and the geometry is only one of those properties. For example, a “Bridge” feature may have a geometry that describes the bridge shape, but also properties such as the maximal weight that it can support, date of last inspection, *etc*. The values of those properties may change with time, for example the maximal supported weight may decrease. However, ISO 19141 does not cover feature properties that change over time. Instead, dynamic or temporal properties are introduced in the OGC® Moving Features Encoding Part I: XML Core (OGC 14-083r2) standard.

Just as a line string is a sequence of points in space, a moving feature is a sequence of events in space and time. Therefore the space and time components can be handled separately — with a spatial CRS (Clause 5) defined for geometries and with dates stored separately in ISO-8601 format such as in the Moving Features CSV encoding (OGC 14-084r2). Alternatively, it is possible to combine spatiotemporal coordinates in a single four-dimensional position. This is allowed by ISO abstract models provided that a four-dimensional CRS can be associated to those positions. Underlying all geometries is the ISO 19107:2003 `DirectPosition` class.

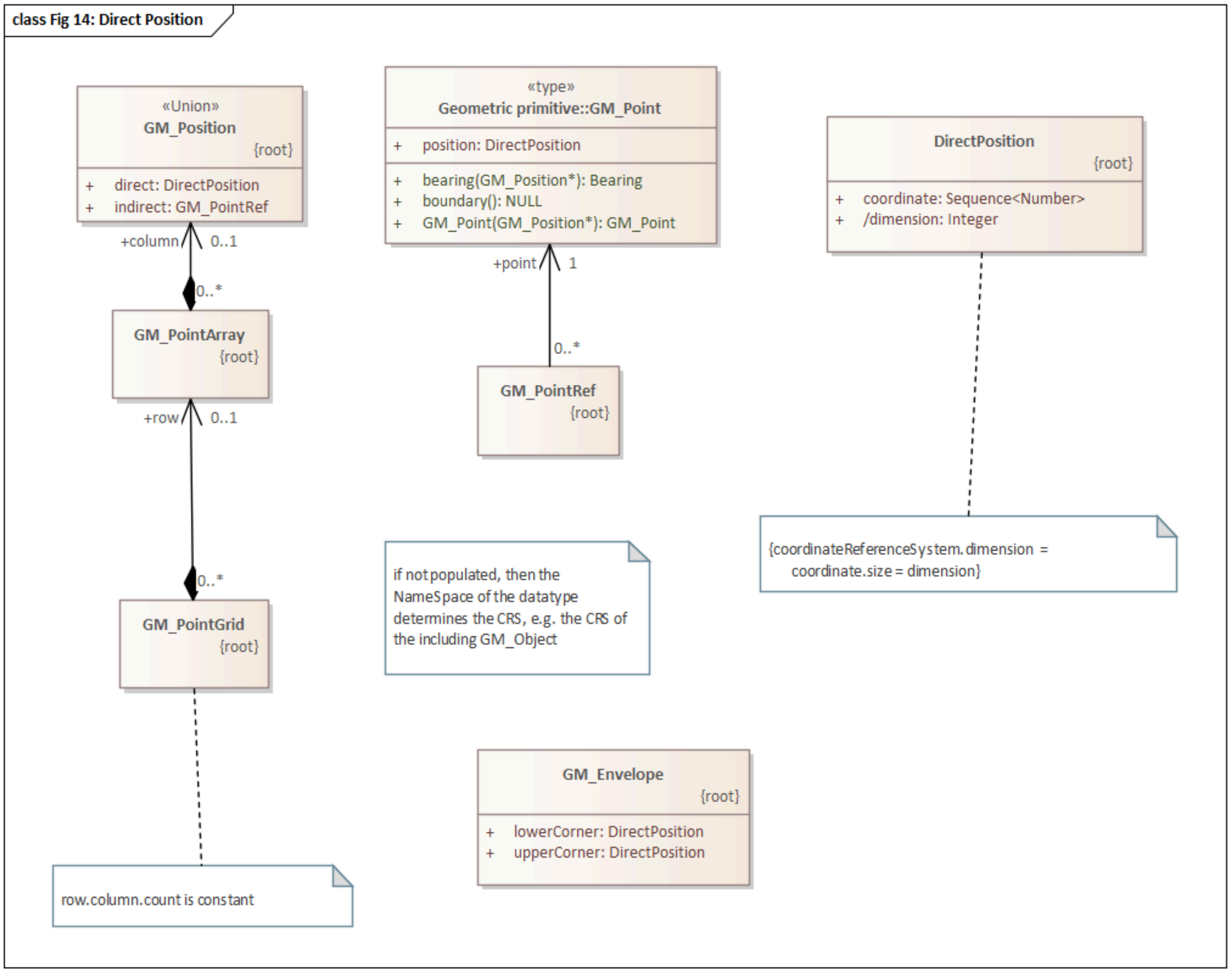
class Fig 14: Direct Position
«Union»
GM_Position
{root}
+ direct: DirectPosition
+ indirect: GM_PointRef
+column
0..1
0..*
GM_PointArray
{root}
+row
0..1
0..*
GM_PointGrid
{root}
row.column.count is constant
«type»
Geometric primitive::GM_Point
+ position: DirectPosition
+ bearing(GM_Position*): Bearing
+ boundary(): NULL
+ GM_Point(GM_Position*): GM_Point
+point
1
0..*
GM_PointRef
{root}
if not populated, then the NameSpace of the datatype determines the CRS, e.g. the CRS of the including GM_Object
GM_Envelope
{root}
+ lowerCorner: DirectPosition
+ upperCorner: DirectPosition
DirectPosition
{root}
+ coordinate: Sequence<Number>
+ /dimension: Integer
{coordinateReferenceSystem.dimension = coordinate.size = dimension}

Figure 3 — UML of position-related classes (source: ISO 19107)

`DirectPosition` has four characteristics:

- defines a coordinate tuple as a sequence of numeric values;
- the number of values in a coordinate tuple is not restricted;
- the `dimension` attribute specifies the number of values in the coordinate tuple; and
- The `dimension` value must be the same as that specified in the coordinate reference system (CRS) associated with the `DirectPosition`.

Therefore, a Feature geometry in four dimensions is allowed if there is a 4D Coordinate Reference System associated with the geometry. The coordinate reference system associated with a `DirectPosition` is defined using the `CRS` class from ISO 19111:2019. Temporal reference systems were initially defined in ISO 19108, but are also included in ISO 19111 since the 2019 revision. The spatial and temporal components can be combined in a single four-dimensional CRS using the `CompoundCRS` class. If spatial and temporal coordinate operations are also defined separately, they can be combined in a single coordinate operation using the `PassThroughOperation` class. Those two ISO 19111 classes are discussed in OGC 22-038 together with examples in Geographic Markup Language (GML). ISO 19107 geometries are described in more details by OGC 22-036 §5.2.4.

The two above-cited approaches are valid when space and time are independent. This is a valid approximation for non-relativist velocities. In situations where Lorentz contractions become significant (Clause 6.2), a third approach based on Minkowski space (Clause 6.3) is required. This is discussed in OGC 22-038.

6.4.1. Non-linearity of time

In its review of theoretical grounds, the Testbed 18 3D+ Standards Framework Engineering Report (OGC 22-036) emphases that very few observed dynamical systems have strictly uniform time units. The transformation from one temporal CRS to another may be a complicated function even if the two CRS definitions seem compatible, for example with the same epoch and same unit of measurement. OGC 22-036 §10.3 calls for a clarification of time scales in ISO 19111. This may be a statement that the `origin` and `calendar` attributes in `TemporalDatum` may not be sufficient for transforming temporal coordinates without the help of an EPSG-like database. This would be consistent with geodetic coordinates, where two reference frames (datums) having different names are considered different datums even if all their attributes, such as `ellipsoid`, have the same values. This is because the `Datum` data structure cannot capture all the real-world complexity (Clause 5.1).

The temporal CRS definitions published in the OGC registry may be misleading in this aspect. For example the following definitions are *“Julian date”* and *“truncated Julian date”* respectively.

- http://www.opengis.net/def/crs/OGC/0/JulianDate
- http://www.opengis.net/def/crs/OGC/0/TruncatedJulianDate

However, the above definitions are currently declared as `TemporalCRS` having their own `TemporalDatum` definitions. By definition coordinate operations between CRSs having different datums are *coordinate transformations*, i.e., a potentially complicated operation with stochastic errors. In reality, *“Truncated Julian date”* is by definition a *“Julian date”* with some digits truncated (more precisely, with an offset simulating a truncation). This is a *coordinate conversion*, an operation with theoretically no error. Coordinate conversions are by definition applied between CRSs sharing the same datum. Consequently *“Truncated Julian date”* should be defined as a `DerivedCRS` having *“Julian date”* as its base CRS. That way, the two temporal CRSs would share the same temporal datum, making clear that the conversion between those two CRSs is really that easy. Without this change, because the above-cited OGC registry has at least one example of datum transformation that should be exact, concluding that this simplicity may be true for many other temporal datums is tempting.

Modifying the definitions contained in the OGC CRS registry for using `DerivedCRS` where applicable would avoid this confusion. However, an obstacle to this change is that OGC definitions are encoded in GML, and the OGC/ISO GML Standard has not been updated to use the latest ISO 19111 revision.

6.4.2. Non-rigid bodies

ISO 19141:2008 and OGC Moving Features XML and CSV encoding standards assume rigid bodies. The case of non-rigid bodies is discussed in both OGC 22-036 and OGC 22-038 and is simplified as follows

- Features are capable of movement in three dimensions. Since these Features have non-trivial three-dimensional shapes which may change over time, the movement of the Feature and the shape of the Feature are two separate properties.
- 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. Example: WGS84 `GeographicCRS`.
- The shape of a Feature is independent of its location. The object's geometry should be self-contained. Therefore, Feature shape should use an internal coordinate reference system. Example: `EngineeringCRS` with the origin at a prominent point in the Feature such as the center of mass.

For rigid bodies, coordinates of Feature geometry in the local CRS are fixed over time. They do not change regardless of any motion by the associated Feature. For non-rigid bodies, those coordinates may change over time and those changes can be described by an ISO 19111 Point Motion operation.

7

STANDARDS

7 STANDARDS

The OGC 22-038 Engineering Report (ER) covers the use of ISO 19111:2019 for defining Coordinate Reference Systems attached to a spacecraft, an observatory platform, the Sun, or Earth (both ECEF and ECI cases). OGC 22-038 provides definition fragments in WKT and GML formats, together with an executable Java application reading a multi-steps chain of operations encoded in GML. The operation steps are as follows.

Voyager spacecraft ⟶ Heliocentric ⟶ ECI ⟶ ECEF ⟶ WGS84 ⟶ Tangent plane

The XML document that describes the above steps has been validated against OGC schema. The demonstration application performs a coordinate transformation using the information provided in the GML document. The parameters specified in the transformation steps are dummy, but the application is a proof of concept showing that existing OGC/ISO standards are already capable of addressing a significant part of extraterrestrial CRS needs. The ISO 19111 shortcomings, such as the absence of `InertialCRS` and `MinkowskiCS`, are summarized in OGC 22-038 and may be relatively minor additions to ISO 19111. Other possible additions are documented in sub-sections of Clause 5 in this ER.

7.1. NASA SPICE

While NASA SPICE is not strictly speaking a standard, it is the software used in practice for coordinate operations between objects in space. SPICE stands for *Spacecraft*, *Planet*, *Instrument*, *Camera-matrix*, *Events*. The SPICE concept was defined by a group of planetary scientists as one result of the 1983 NASA-mandated Planetary Data Workshop. This section summarizes some SPICE concepts and how they relate to ISO 19111.

7.1.1. Reference Frame

An ordered set of mutually orthogonal, unit length direction vectors, and possibly time dependent vectors. The frame has an associated center which must be the frame's origin (0,0,0). When working with a body-fixed frame, this is the center of a natural body such as the sun, a planet, a satellite, a comet, or an asteroid. The body center location is specified in SPK files (SPICES proprietary software kit). SPK files also house the center location of topocentric, instruments, or spacecraft frames.

ISO 19111 equivalences: `Datum` and its subclasses (`GeodeticReferenceFrame`, `TemporalDatum`, *etc.*), except that vectors are not necessarily mutually orthogonal in the ISO standard. Furthermore, the center (anchor) is specified only by a free text in ISO 19111. However, location may be specified indirectly, relative to another frame, by a *late-binding* coordinate transformation.

7.1.2. Inertial reference frame

A non-rotating (relative to distant stars) reference frame. The velocity may be non-zero, but the acceleration is negligible. SPICE uses the J2000 frame, which was considered the best realization until the development of ICRF (International Celestial Reference Frame). Transformation from J2000 to Earth Fixed frame uses the 1976 IAU Theory of Precession, the 1980 Nutation model, and the Greenwich Mean apparent Sidereal Time.

Source: Ansys Government Initiatives (AGI)

ISO 19111 equivalences: None. Clause 5.3 proposes adding `InertialReferenceFrame` and `InertialCRS` classes to the abstract model.

7.1.3. Non-inertial reference frame

Accelerating frames, including accelerating by rotation. Candidates for utilization of this frame include the following.

- **Body-fixed:** associated with a natural body such as planets and satellites
- **Topocentric:** associated with objects near the surface or on the surface of a natural body
- **Spacecraft and instrument frames:** associated with objects that are attached to a spacecraft such as solar arrays, scan platforms, instruments, scanning mirrors and antennas
- **Instrument:** one or several frames can be associated with each instrument (applicable to a solar array and spacecraft antenna)

ISO 19111 equivalences:

- `GeodeticReferenceFrame` for the body-fixed case
- `DerivedCRS` for the topocentric case, with the geodetic reference frame as the base CRS
- `EngineeringCRS` for the spacecraft case
- `DerivedCRS` for the instrument case, with the spacecraft CRS as the base CRS

7.1.4. Dynamic reference frame

A frame with a time-dependent orientation. This category does not include frames for which the orientation is provided using a spacecraft orientation or a natural body orientation.

ISO 19111 equivalences: No direct equivalence, but the same functionality can be achieved with time-varying transformation parameters. See the transformation step from ECI to ECEF in the OGC 22-038 ER. Note that ISO 19111 also has its own concept of “dynamic reference frame”, but it is used for a different problem.

7.1.5. Coordinate System

A mechanism for locating points within a reference frame. SPICE uses right-handed Cartesian coordinate systems.

ISO 19111 equivalences: `CoordinateSystem` and its subclasses (`CartesianCS`, `SphericalCS`, *etc.*).

7.1.6. Planetocentric coordinate system

A system where longitude is measured positive eastward, and latitude is the angle between the equatorial plane and a line from the center of the body to a given point on surface.

ISO 19111 equivalences: `GeodeticCRS` associated to a `SphericalCS`.

7.1.7. Planetodetic Coordinate System

A system where longitude is the same as with planetocentric, but where latitude is the angle measured from the equatorial plane and a line perpendicular to the surface at the point of interest.

ISO 19111 equivalences: `GeodeticCRS` associated to an `EllipsoidalCS`. The `GeographicCRS` specialization should be used.

7.1.8. State

An understanding of the reference frame and coordinate system that is going to be used. A state may be defined by position and velocity.

ISO 19111 equivalences: `CoordinateMetadata`. If there is a need for more metadata than what ISO 19111 provides, the `CoordinateMetadata` class can be extended.

7.2. GeoPose

GeoPose is an OGC standard for defining reference frames anchored to the earth’s surface or within other astronomical coordinate systems. The anchor is called the “outer frame” and is defined by a CRS. The other frames are called “inner frames” and are related to the outer frame by a chain of operations. The way that inner frames are defined in GeoPose is conceptually a

`DerivedCRS` in ISO 19111, with only a different encoding compared to WKT or GML formats. For example, consider the following GeoPose snippet:

```
{
  "validTime": 1674767748003,
  "poseID": "https://carlsmyth.com",
  "parentPoseID": "https://example.com/nodes/MarsExpress/1/Wagons/3",
  "frameSpecification":
  {
    "authority": "https://ogc.org"",
    "id": "ENGINEERINGCRS[…snip…]",
    "parameters": "{\"x\": 1234567890.9876,\"y\": 2345678901.8765, \"z\":
3456789012.7654}"
  },
  "quaternion":
  {
    "x":0.0174509,"y":0.0130876,"z":-0.0002284,"w":0.9997621
  }
}
```

Figure 4 — GeoPose snippet

NOTE“Authority” and “ID” attributes should be used for identifying an entry in a registry such as the EPSG geodetic registry or the OGC definitions server. But in the above snippet, `"authority": "https://ogc.org"` means that `id` is a CRS standalone definition given in WKT format. It may be less confusing to replace the `attribute` and `id` pair by a different attribute such as `wkt` for those cases, especially since WKT is not the only CRS encoding that exists in OGC Standards.

The above snippet could be as below in ISO 19162 WKT format. For making the WKT more compact, this example uses the fact that `UNIT` elements in `PARAMETER` and `ORDER` elements in `AXIS` are optional.

```
ENGINEERINGCRS["Carl Smyth"
    BASEENGCRS["Mars Express Wagon 3", …snip…],
    DERIVINGCONVERSION["Carl position",
        METHOD["quaternion"],
        PARAMETER["x",  0.0174509],
        PARAMETER["y",  0.0130876],
        PARAMETER["z", -0.0002284],
        PARAMETER["w",  0.9997621],
        PARAMETER["valid time", 1674767748003, UNIT["second", 1]]],
    CS[Cartesian, 3],
    AXIS["ahead (x)", forward],
    AXIS["right (y)", starboard],
    AXIS["down (z)", down],
    ID["The voyager company", "Carl Smyth"]]
```

Figure 5 — WKT equivalent for GeoPose snippet

The GeoPose Standard could be reformulated in terms of ISO 19111. For example, it would be possible to state that a GeoPose encoding is a shorthand for an ISO 19111 `DerivedCRS` where the coordinate system is implicitly Cartesian, and some keywords are omitted (e.g., `"x":0.0174509` instead of `PARAMETER["x", 0.0174509]`), *etc.* However, the underlying abstract model could still be the ISO 19111 one.

Defining a full CRS at each step may seem unnecessarily complex. But a CRS is the combination of a reference frame (datum) and a coordinate system. In a `DerivedCRS`, the datum is inherited from the base CRS and does not need to be repeated. The remaining part to define is the coordinate system. That part is not necessarily a repetition, because ISO 19111 gives a choice between different kinds of coordinate systems to better match the feature geometry (Clause 5.2).

7.2.1. GeoPose early-binding

A more fundamental issue with GeoPose is that it is an *early-binding* model. The coordinate operation that relates an inner frame to its outer frame is bundled in the frame definition itself. Consider three Coordinate Reference Systems denoted *A*, *B*, and *C*. Early binding models are based on the logic that if a conversion $A \rightarrow B$ exists together with a conversion $A \rightarrow C$, then a conversion $B \rightarrow C$ can be inferred as $(A \rightarrow B)^{-1}$ followed by $(A \rightarrow C)$, or said otherwise $(B \rightarrow A)(A \rightarrow C)$. This is true when *conversions* are exact, but this is not accurate with *transformations* in the geodesic world (see ISO 19111 distinction between “conversion” and “transformation”). Another way to see the problem is to note that *“if A=B and B=C then A=C”* seems universal logic, but that statement is true only if the `=` operator represents strict equality. When that assumption is not true, then $A \approx B$ and $B \approx C$ does not necessarily imply $A \approx C$. Clause 8 illustrates the problem with a use case.

Another way to describe the problem is to note that early-binding models tend to encourage the use of a “universal CRS”, which was frame *A* in above $(B \rightarrow A)(A \rightarrow C)$ scenario. In GeoPose, the “universal CRS” is the root of a GeoPose graph. In Well-Known Text (WKT) version 1 (OGC 01-009), the “universal CRS” was WGS84 and the relationship of any CRS to that “universal” one was encoded in the `TOWGS84` keyword. That keyword is deprecated and removed from ISO 19162 (a.k.a. WKT 2). Annex A gives a concrete example where `TOWGS84` does not work. Another example is shown in the figure below derived from EPSG geodetic dataset version 8.9.2. There is no direct relationship between WGS84 and Japanese Geodetic Datum 2011. Note also that there are 85 different transformations between NAD27 and WGS84 (see Figure 9 for more explanation).

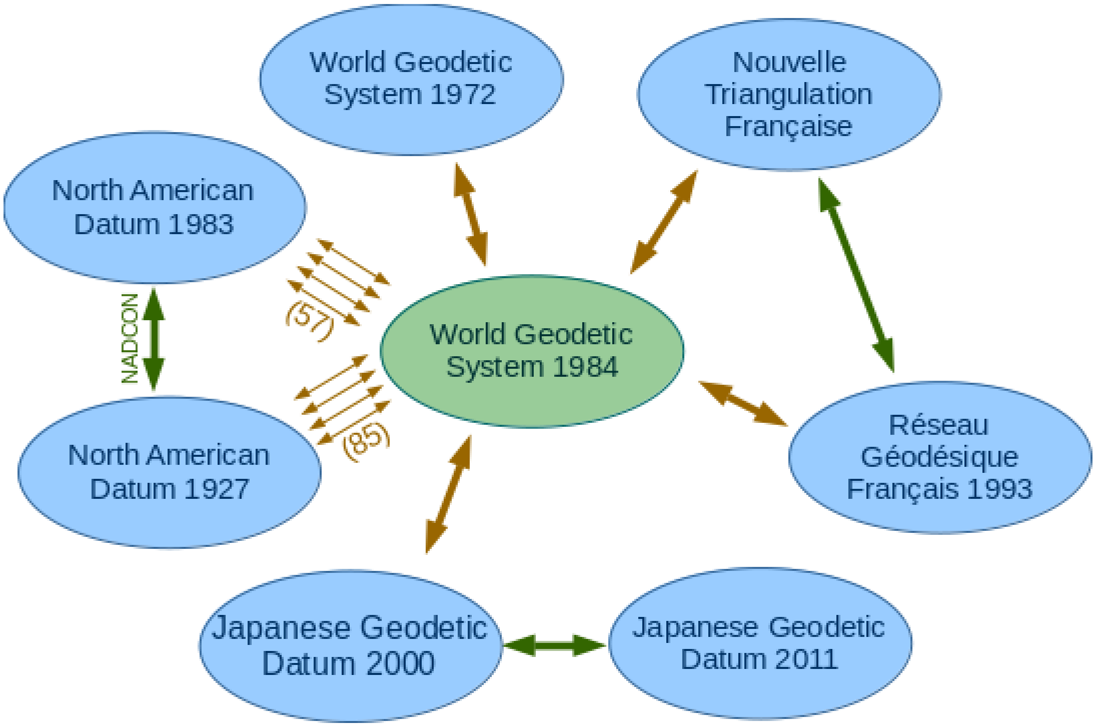
World Geodetic System 1972
Nouvelle Triangulation Française
North American Datum 1983
NADCON
(57)
World Geodetic System 1984
North American Datum 1927
(85)
Réseau Géodésique Français 1993
Japanese Geodetic Datum 2000
Japanese Geodetic Datum 2011

Figure 6 — Example of transformations in EPSG geodetic dataset version 8.9.2

Allowing GeoPose to specify full CRS definitions at arbitrary steps does not solve the problem. The crux of the problem is that $(B \rightarrow A)(A \rightarrow C) \overset{?}{=} (B \rightarrow C)$ is not always true. Consequently when defining frame C, it is not sufficient to provide the $A \rightarrow C$ transformation. The frame definition would also need to provide the $B \rightarrow C$ transformation, as well as the transformations between all other pairs of CRS involving C. The set of transformations can be reduced by excluding those that are not more accurate than the indirect path. This approach is still hardly extensible. If a frame D is added later, it should not force an update of frame C for adding the $D \rightarrow C$ transformation. It is easier to add/remove rows in a separated table of coordinate transformations, without changing the frame definitions. Especially since in software implementations, the recommendation is that frames and CRSs are immutable objects.

8

IMPORTANCE OF DATUM

8 IMPORTANCE OF DATUM

In this section, a use case is provided to help explain why datum information is important. From the importance of datum information, the ISO 19111 distinction between *transformation* and *conversion* is significant since the former implies a datum change while the latter is restricted to operations that do not change the datum. From the distinction between conversions and transformations, a distinction between the cases where an *early-binding* model can be used (i.e., when using conversions) and when a *late-binding* model should be used (i.e., when using transformations).

The use case is as follows. A space agency on Earth has determined that a small piece of space debris is going to collide with a station in orbit. The location of the collision is predicted in the reference frame used by Earth observatories, for example WGS84. After the collision happens, the exact location is observed by the crew onboard the space station. Then another spacecraft is sent to the station to perform repairs.

For the sake of simplicity, this ER section considers only the uncertainties caused by datum changes. All other uncertainties are ignored. This is emphasized by texts such as "assuming that *Foo* has infinite precision." In the real world, the limited accuracy of *Foo* would be taken in account by combining its uncertainty with the datum change uncertainty. But doing so does not change the purpose of this section, which is to explain the importance of datum information.

For illustrating the uncertainty discussed in this section, consider a blue reference frame and a red reference frame. Each reference frame has its own origin, illustrated by an anchor in the following figures. In each frame, the location of 4 or 5 features are given. All coordinates in the blue frame are relative to the origin of that frame. Assuming an infinite precision, feature locations in the blue frame are represented by points.

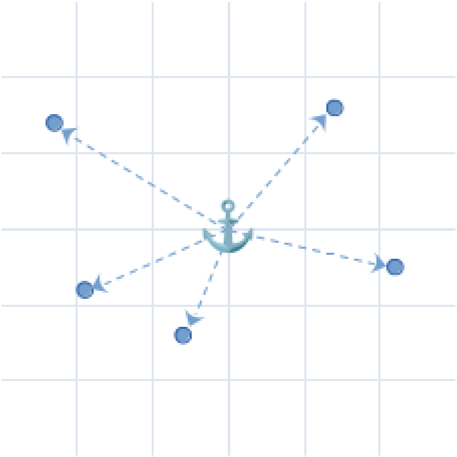

Figure 7 — Locations observed from the blue frame

Likewise, all coordinates in the red frame are relative to the origin of that frame. Assuming an infinite precision, feature locations in the red frame are also represented by points.

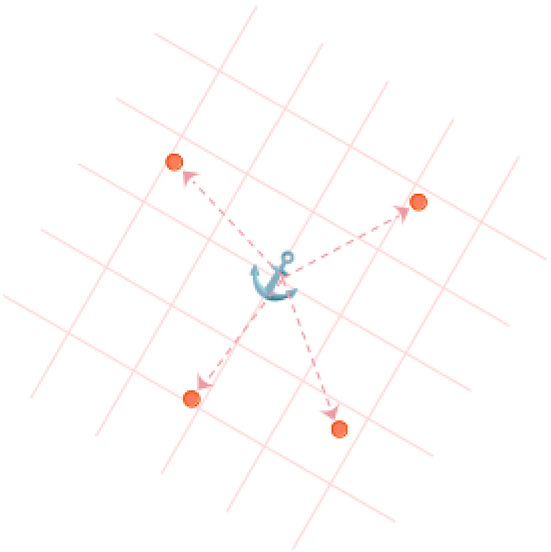

Figure 8 — Locations observed from the red frame

Now if a process needs coordinates of all features in a single frame, there is a choice: either transform coordinates from the blue frame to the red frame, or conversely. Those two choices are not equivalent if the transformation between the two frames does not have infinite precision (ignoring rounding errors and other limitations of floating point arithmetic). Coordinate transformation inaccuracies are caused by stochastic errors, not by software limitations.
The origin of one frame may not be well defined from the perspective of another frame. For example, because the origins of both frames are not within sight of each other. Before the satellite age, this could be because the datums were separated by a large water body or a mountain chain. This uncertainty is represented by blurred anchors in Figure 10 and Figure 11.
The transformation can also be irregular, requiring interpolations in datum shift grids. This is the case of NADCON transformation for example, as shown below.

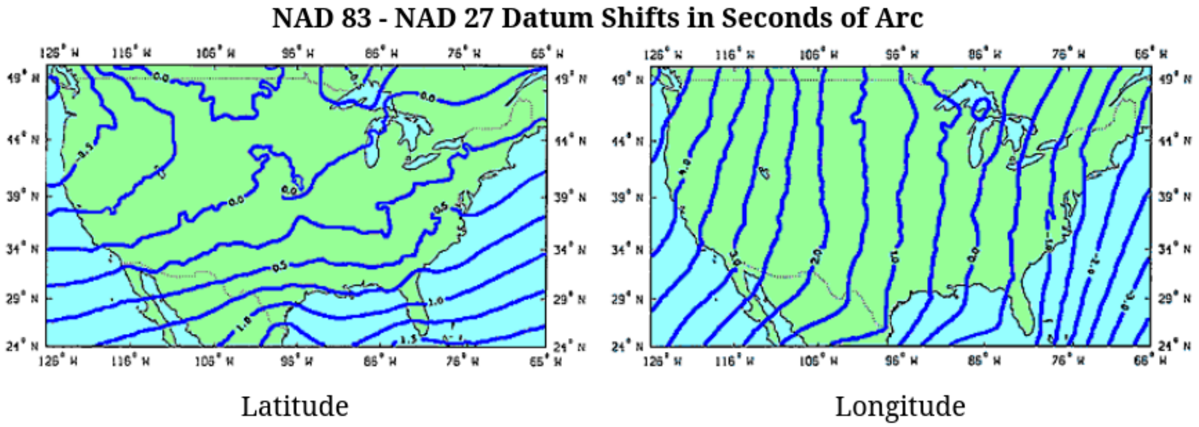

Figure 9 — Source: NOAA National Geodetic Survey (NGS)

Alternatively, if a transformation is considered too complex for the desired accuracy, approximations can be used. For example, the above NADCON transformation can be replaced by a Helmert transformation (a combination of scale, rotation, and translation in geocentric coordinates). This simplification replaces the thousands of values in NADCON grids by seven parameters. Consequently, many transformations may exist between the same pair of CRSs. There are various compromises between complexity and accuracy, with different domains of validity. Indeed, the EPSG database defines approximately 85 different transformations between the same pair of CRSs (NAD27 and WGS84 in Figure 6) which are different NADCON approximations valid for different US states. These approximations can be understood as linear regressions computed using different subsets of the NADCON data illustrated in Figure 9.

If a process performs its calculation in the red reference frame, then coordinates in the blue frame must be transformed to coordinates in the red frame. But because of the transformation uncertainty, locations of blue points relative to the red frame's origin are uncertain while the locations of red points keep their accuracy.

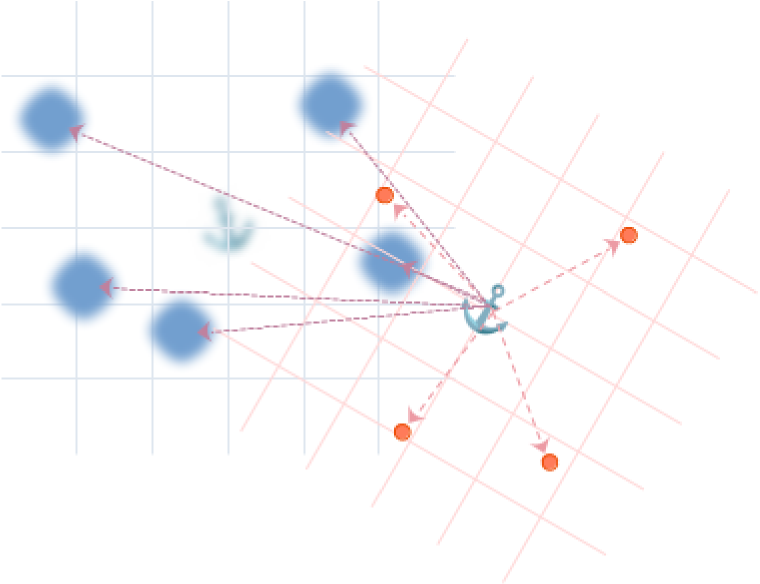

Figure 10 — Locations observed from the red frame

Conversely, if the process rather performs its calculations in the blue reference frame, then the set of points that keep their accuracy and the set of points that become uncertain are the opposite of previous example.

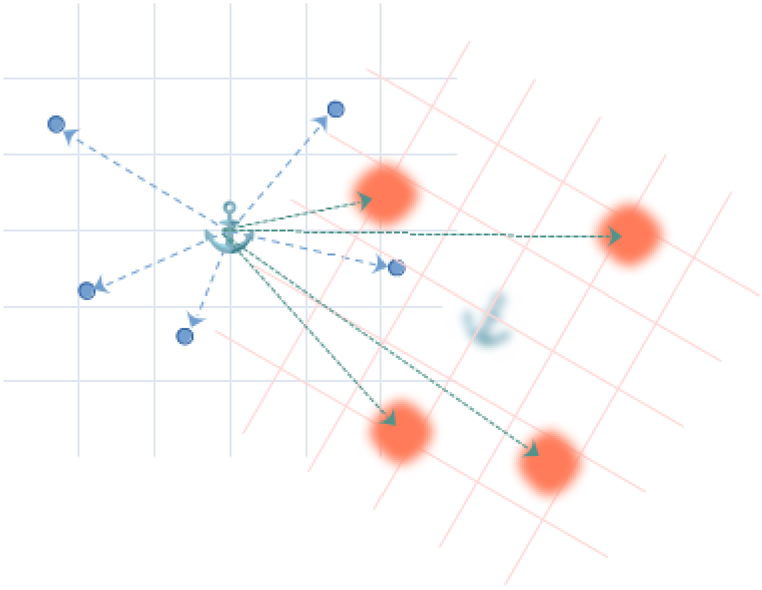

Figure 11 — Locations observed from the blue frame

So even if at this stage, all points are located at the same places in both of the above figures, their uncertainties are very different depending on which datum has been chosen. This information is stored in the target Coordinate Reference System. In the next section discussion will show that not only the uncertainties are different, but the transformation results become different when more transformation steps are chained after the transformations illustrated above.

8.1. Debris collision scenario

Let’s assume a small piece of space debris with a trajectory which is perfectly predicted by a numerical model working in the WGS84 reference frame. In reality the predicted trajectory would be uncertain. But that uncertainty is ignored in this section by focusing on the datum changes. This is represented by a clear line from Earth center to the space debris below, as opposed to the blur line from Earth center to the space station where a datum change will occur.

The space debris may be located using spherical coordinates (r, θ, φ) which will need to be converted to Cartesian coordinates (X, Y, Z). If the latter are geocentric coordinates relative to the same WGS84 datum, then this is a *conversion*, not a transformation. A conversion has no precision lost (ignoring rounding errors): the location in new Cartesian coordinate system is as accurate as the location in the previous spherical coordinate system. In ISO 19111, such conversions can be part of a `DerivedCRS`.

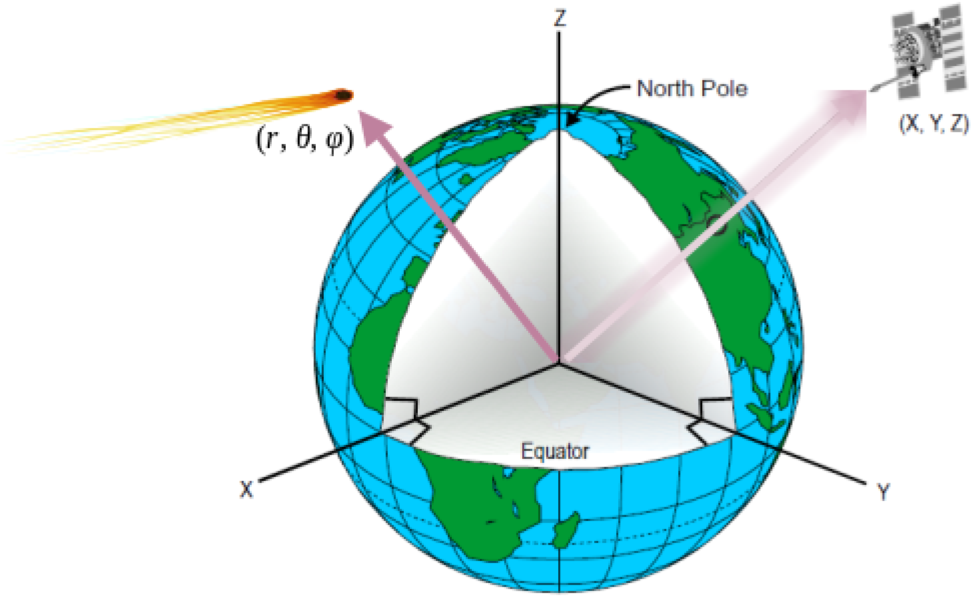

Figure 12 — Debris and space station viewed from WGS84

Assume a space station is going to be hit by the space debris. The space agency wants to predict which part of the station will be hit. The hit coordinates need to be expressed in the local coordinate reference system of the station, which has its origin at the station center of mass. This implies that the (*X*, *Y*, *Z*) coordinates relative to WGS84 center of mass need to be transformed to (*x*, *y*, *z*) coordinates relative to the space station center of mass. Let's assume that this transformation is not perfectly determined. This may be, for example, because the distance between Earth and the space station is measured by instruments affected by some kind of noise. From the WGS84 perspective, the debris location is well defined but the station location has some uncertainty as illustrated below.

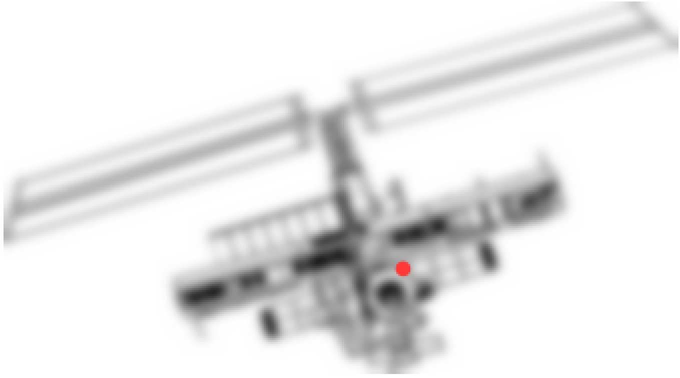

Figure 13 — Predicted impact location viewed from WGS84

From the station perspective, it is the location of WGS84 center of mass which is uncertain. Consequently, the debris location, which is relative to that uncertain WGS84, is also uncertain.

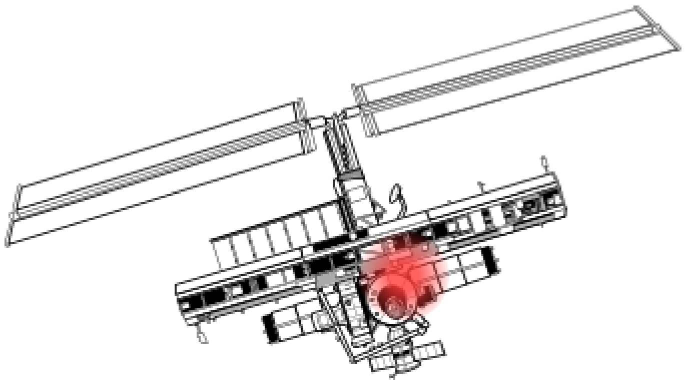

Figure 14 — Predicted impact location viewed from the station engineering CRS

Now assume that the impact happened. Occupants of the space station can observe the exact hit location in the coordinates of the local CRS. Those coordinates are transmitted to another spacecraft which will come to make repairs. However, that other spacecraft needs to move its instruments according its own local CRS. Consequently, the impact coordinates need to be transformed to the spacecraft local coordinate system. There are two choices:

- transform the predicted hit location, which is relative to WGS84; or
- transform the observed hit location, which is relative to space station local CRS.

NOTEIn the introduction to this theoretical scenario, the assumption was that the debris trajectory was predicted with infinite precision in WGS84. This is of course unrealistic, but the intention is to ignore positioning errors and focus on transformation errors. In this theoretical scenario, the hit location is perfectly known in the two reference systems. The fact that this information could be used for accurately defining the transformation between the two reference systems is ignored.

Now assume that the transformation from WGS84 to the spacecraft local frame is determined by observations from the ground as illustrated by the antenna below. In this case assume that this transformation has uncertainties in the same way that the transformation from WGS84 to the space station has uncertainties. The use of WGS84 as a universal hub leads to the transformation path shown below.

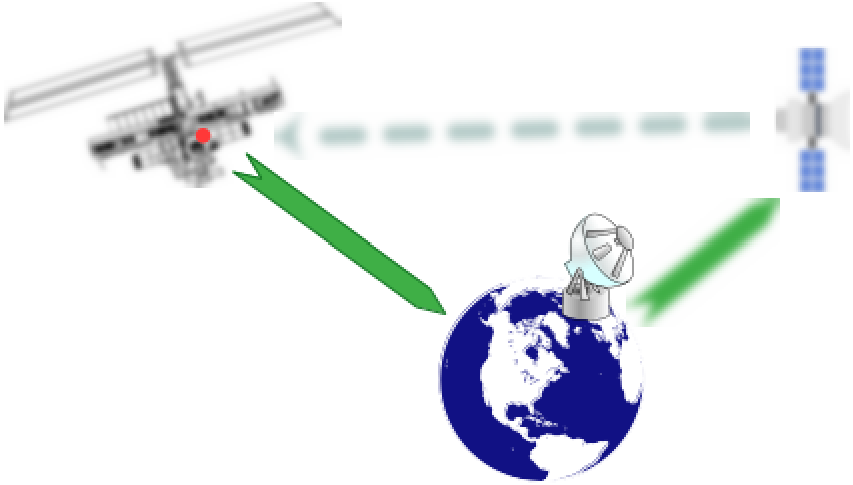

Figure 15 — Hit location transformed from WGS84 frame

As stated previously, the hit coordinates in this scenario are assumed to be perfectly known in the WGS84 reference frame, so the green arrow on the left side is not blurred (to focus on the uncertainty of one transform at a time). In a more realistic scenario, which would use the hit coordinates observed by the station crew in their local reference system, the left arrow would need to be blurred as well and the uncertainties of both transforms would need to be combined.

Finally, assume that the repair spacecraft has radar and can compute its own transformation between its reference frame and the station reference frame. While not perfect (the arrow is still blurred), this transformation may be more accurate than the one passing through WGS84, for example, because both platforms are close to each other.

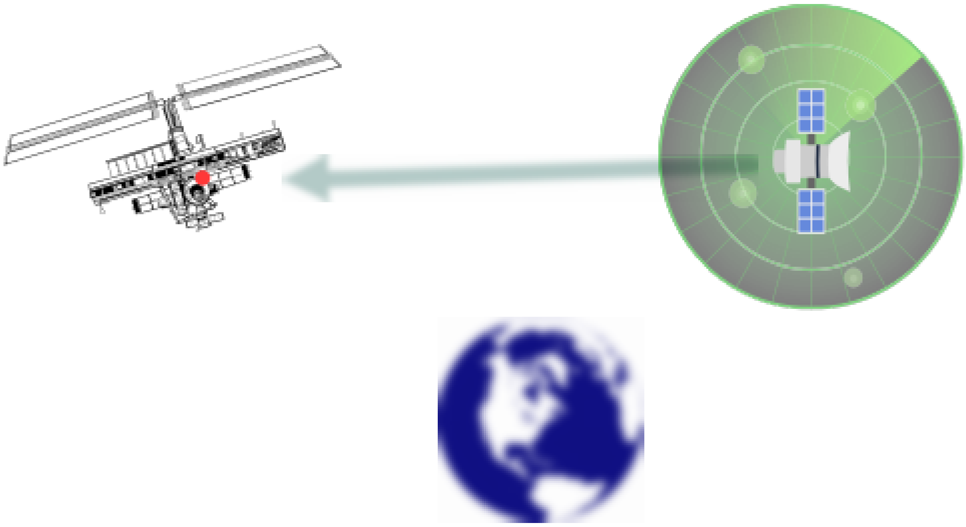

Figure 16 — Hit location transformed from space station local frame

Because the transforms between the repair spacecraft and the space station were computed using different data (different radars) in the two above figures, those coordinate transformations will produce slightly different results. Consequently, even in the (unrealistic) scenario where the coordinates of an event are known with infinite precision in two different reference frames (datums), the datum of the point selected as input for the transformation will have an incidence on the numerical values of the output coordinates.

In practice, the input points will not have infinite precision. In a more realistic scenario where the predicted impact location is uncertain, there is a compromise between accepting the predicted coordinates with their uncertainty, or using the coordinates observed by the crew with the uncertainty added by the coordinate transformation (if WGS84 coordinates are desired). The best choice may seem obvious in the scenario presented in this section, but it may not be so obvious for frames far away, when data are rare or when instruments are damaged. The definition of a chain of coordinate operations should be clear about which datum are used, and where datum changes are occurring. The ISO 19111 model supports that goal.

9

KEY TAKEAWAYS

9 KEY TAKEAWAYS

The ubiquity of the EPSG geodetic dataset on Earth makes it easy to forget that it is at play during coordinate operations (Clause 5.1). An authority equivalent to EPSG will be needed for defining not only reference frames in space, but also the *late-binding* (Annex A) transformation parameters.

Some notes about possible extensions to existing OGC and ISO Standards are as follows.

- Having an attribute in `CartesianCS` stating whether the coordinate system is right-handed may be useful (Clause 5.2.5).
- Having an attribute in `AffineCS` giving the angles between pairs of axes may be useful (Clause 5.2.4).
- `EllipsoidalCSs` cannot be associated to `EngineeringCRSs`. Whether this is a problem or not has not been determined (Clause 5.2.1).
- The fact that the GML Standard has not been updated to the latest ISO 19111 revision will be an obstacle to the definition of some CRS's.

Perhaps ISO 19111 should emphasize that two datums with different names may be related by complicated transformations even if all their attributes (`ellipsoid`, `origin`, `calendar`, …) have identical values. This complication applies to temporal datum as well as geodetic reference frames. The risk of confusion is illustrated by the CRS definitions published in http://www.opengis.net/def/crs/OGC/0/. This registry declares *"Julian date"* and *"truncated Julian date"* as two CRSs with two different datums, while actually the latter should be a `DerivedCRS` sharing the datum of the former (Clause 6.4.1). This confusion exists in the implementation of Apache SIS version 1.3 as well.

The concepts developed in GeoPose Standard and in NASA SPICE software can be mapped to ISO 19111 concepts. The ISO 19111 Standard appears to have the richest set of capabilities for expressing complex chain of coordinate operations while taking into account constraints of the physical world such as the uncertainties caused by changes of datum (Clause 8).

A

ANNEX A (INFORMATIVE) EARLY BINDING VERSUS LATE BINDING

A

ANNEX A (INFORMATIVE) EARLY BINDING VERSUS LATE BINDING

Consider a coordinate transformation from Martinique 1938 (EPSG:4625) to RGAF09 (EPSG:5489). In an early-binding approach, the `TOWGS84` parameters associated to reference frames would be as follows.

Table A.1 — Early-binding parameters

CRS	TX	TY	TZ	RX	RY	RZ	ΔS
Martinique 1938 *(source CRS)*	186	482	151				
RGAF09 *(target CRS)*	0	0	0				

The parameters for the transformation from source CRS to target CRS can be estimated by (parameters of target CRS) − (parameters of source CRS). However, this is a concatenation of two transformation steps where each step is an approximation. Chaining two approximations is not the same as performing a single approximation for the full chain. In the following table, the “early-binding” row shows the result of the above chain of two approximations, while the “late-binding” row shows the result of a single approximation as reported in EPSG database.

Table A.2 — Martinique 1938 to RGAF09 transformation parameters

METHOD	TX	TY	TZ	RX	RY	RZ	ΔS
Early-binding	186	482	151				
Late-binding	127.744	547.069	118.359	-3.1116	4.9509	-0.8837	14.1012

The consequence is an error of about 1 meter for the early-binding method, while the expected accuracy for late-binding method is about 0.1 meter. There is no way an *early-binding* implementation can guess with certainty the best parameters when none of the source and target CRSs are WGS84.

B

ANNEX B (INFORMATIVE) PRECISE POSITIONING

B

ANNEX B (INFORMATIVE) PRECISE POSITIONING

B.1. Network Time Protocol

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 of up to 2-64 seconds. However, precision is not accuracy. Processing the NTP packets takes time. It also takes time for the packets to traverse the network. Therefore, 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 the final timestamp is the time of the response packet reception. Therefore the four times stamps are:

- t0 – request packet transmission timestamp;
- t1 – request packet reception timestamp;
- t2 – response packet transmission timestamp; and
- t3 – 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:

$$\theta = ((t_1 - t_0) + (t_2 - t_3)) / 2 \tag{B.1}$$

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:

$$\delta = (t_3 - t_0) - (t_2 - t_1) \tag{B.2}$$

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 be 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 to stratum 15. Lower stratums naturally lead to less accuracy. Note: The time server includes its stratum level in the NTP response packet.

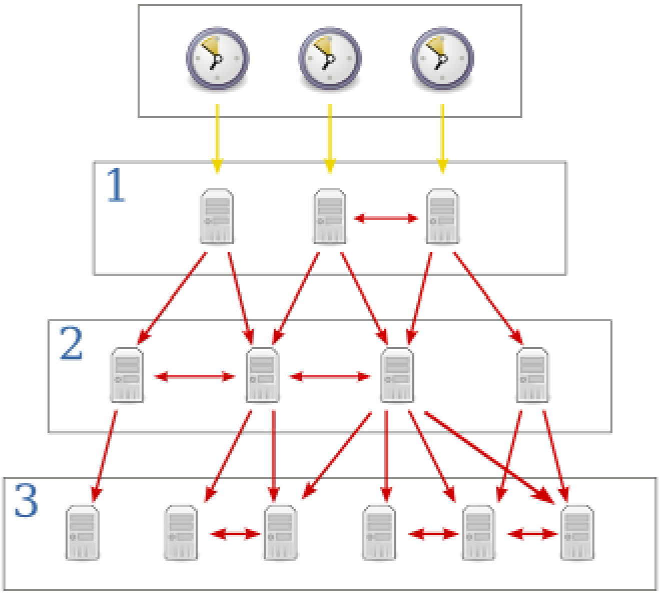

Figure B.1 — Network Time Protocol servers and clients

B.2. 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:

$$\rho = c(\Delta t / 2) \qquad \text{(B.3)}$$

Where:

- ρ = the measured distance<
- c = the speed of light
- Δt = the interval between emission of the laser pulse and detection of its reflection

The figure below 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.

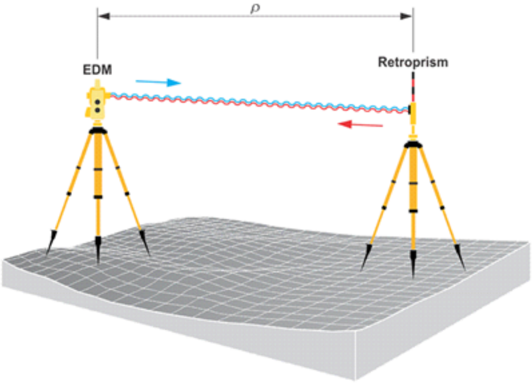

Figure B.2 — Source: GPS for Land Surveyors

There is an issue though: with the speed of light at 3×10^8 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. Such accuracy is 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 the offset between the transmitted and received signal can be measured, then the time delay can be calculated. The wavelength of a signal can be derived from the frequency using the formula:

$$\lambda = c / h \tag{B.4}$$

Where:

- h = frequency
- λ = wavelength
- c = the speed of light

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

$$P = (N\lambda + d)/2 \tag{B.5}$$

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 below). While high-precision clocks are difficult to build, a tuning circuit capable of measuring the phase shift is simple and inexpensive.

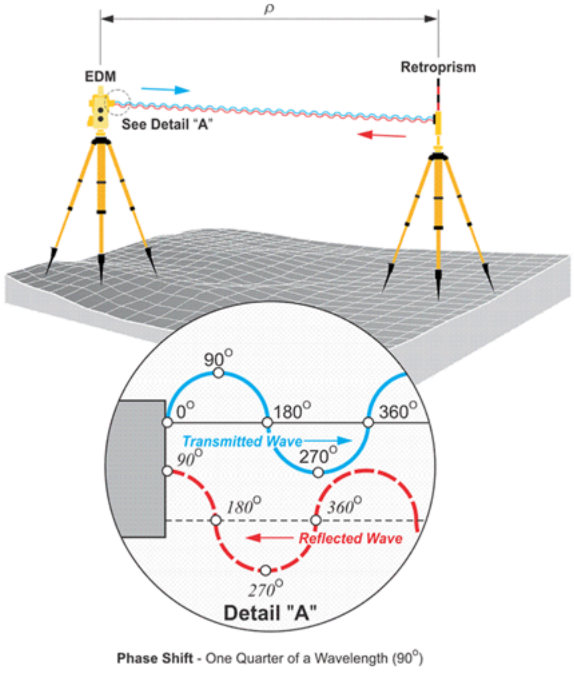

Figure B.3 — Source: GPS for Land Surveyors

A difficulty is to 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, 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 when the signal was transmitted is known, 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:

$$N = \Delta t \times h \tag{B.6}$$

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

B.3. GPS

The Global Positioning System (GPS) is the most widely known precise positioning technology used today. Yet, the GPS satellites orbit 20,183 km above the earth surface. This section describes how something so far away can provide measurements so precise.

B.3.1. 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 the US Naval Observatory (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; and
4. the acts of transmitting the NAV messages shall be executed on SV time.

B.3.2. GPS Signal

GPS signals are driven by the on-board atomic clocks. Four frequency bands are used.

Table B.1 — Frequency bands used by GPS

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. A carrier serves as a medium upon which other signals can be superimposed. This is the same principle as an FM radio. A radio is tuned to a carrier frequency. The sound emitted by the radio 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 as follows.

- **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. The NAV 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.

B.3.3. 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 below). 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 a sphere is constructed around each control point (P_i) of radius (L_i), then the location of *A* is at the intersection of the three spheres.

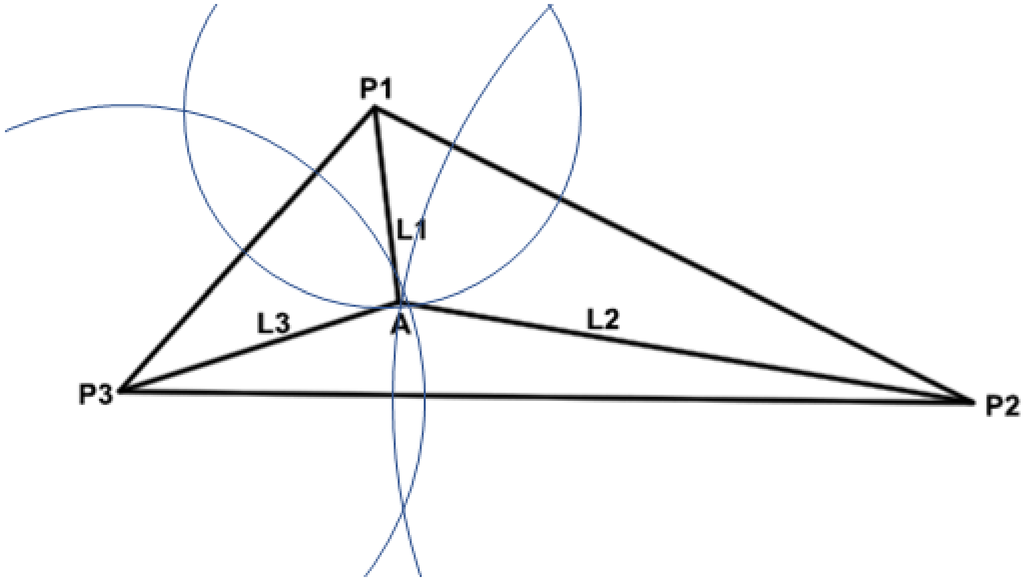

Figure B.4 — Trilateration

Therefore, all that is required for satellite-based precise positioning is the locations of P_i and the distances L_i. This is 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 nanosecond (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. Therefore, the receiver first calculates the pseudorange observable and iterates to find an accurate solution. The pseudorange is *an approximation of the distance between a satellite and a GNSS receiver*.

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:

$$P_i = c(t_a - t_i) \qquad \text{(B.7)}$$

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.

$$P_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} + c(\tau) - c(\tau_i) \qquad \text{(B.8)}$$

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.

The satellite location and clock bias are provided through the NAV message. This leaves four unknowns to be considered. The location of *A* (x, y, z) and the receiver clock bias (τ). If working with four satellites, then that gives 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.

B.3.4. 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:

$$P_i = N_i \times \lambda + d_i \qquad \text{(B.9)}$$

Where:

- N_i = the number of full wavelengths received at the detector
- λ = the wavelength of the carrier
- d_i = the fractional part of a wavelength received

The carrier phase observation can also be represented using the Carrier Phase Bias (B_i) where:

$$B_i = \lambda(\backslash Phi - \backslash Phi_i - N_i) \qquad \text{(B.10)}$$

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:

$$P_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} + c \times \tau - c \times \tau_i + B_i \qquad \text{(B.11)}$$

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

B.4. Moving Features and Sensor Integration Introduction

Moving Features is a vital part of the future of the Internet of Things (IoT). The fast-paced growth of digital motion imagery and advancements in machine learning technology continue to accelerate widespread use of moving feature detection and analysis systems. Moving Features can be any moving Feature of Interest (FOI) that can be tracked using multiple sensors and may include static features and observations that allow for easier tracking and provide information about the feature. This data can then be refined and used by a series of software programs to provide a searchable Client that an end-user can access to choose a Feature and observe the observation.

Ingestion services take in raw data, give the feature an ID, and feed the data and IDs to the Sensor Hub. The Sensor Hub collects data from one or multiple ingestion services, harmonizes it into a common data model, and integrates them into a data bank that can be accessed by the Client. The Client can browse the data after searching for a feature that matches the ID originally given the information by the ingestion service.

Celestial bodies can be tracked using a similar moving features system as described in the Testbed-18 (TB-18) Moving Features and Sensor Integration (MFSI) task. In the TB-18 MFSI task, ships and hurricanes were tracked as a collection of moving features using data from the National Oceanic and Atmospheric Administration (NOAA) that fed into a Sensor Hub where the data was harmonized into a common data model and stored. The Client would then retrieve that stored data when prompted by the user and a refined searchable set of data could be used. The moving feature ingestion service could obtain data from, for example, the Sloan Digital Sky Survey used by the U.S. Naval Observatory, Princeton University, and the University of Chicago, and convert that data using a data-processing pipeline.

A pipeline is a software program that processes digitized data automatically to extract certain types of information. Another potential data source would be data from the U.S. Tracking and Data Relay Satellite System (TDRSS) that tracks satellites and uses data from the Multi-Mission Flight Dynamics lab that helps track spacecraft as well as known celestial objects, such as the sun and moon. With this information, users and/or applications can track specific Features of Interest (FOI), such as a planet or star, and Observations associated with the feature. The ingestion service could then post the FOIs and observation data to the Sensor Hub where multi-source datasets (planet movements, meteor tracks, etc.) are stored and interpreted as moving features associated with static features. The Sensor Hub would then be an aggregation layer that collects the data from one or multiple sources and integrates into a common data model. On the ingestion side, the Sensor Hub implements the draft OGC Connected Systems API standard. This draft API standard specifies how to accept feature descriptions in GeoJSON format and observations in the O&M-JSON format. On the consumer side, a client can either use an implementation instance of the Connected Systems API standard to browse through features of interest and observations or view the same data in the MF-JSON format via the draft “OGC API — Moving Features” (MF API) interface. The Client then retrieves the data using AI to refine the data further and displaying the data in orthorectified images with harmonized resolution and metadata easy to understand by the user.

C

ANNEX C (INFORMATIVE) REVISION HISTORY

C

ANNEX C (INFORMATIVE) REVISION HISTORY

DATE	RELEASE	AUTHOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
2022-07-12	0.1	Logan Stark. Editor	All	initial version
2022-11-04	0.2	Chuck Heazel	Precise positioning	initial version
2023-02-01	0.3	Martin Desruisseaux	All	Edition or formatting
2023-02-06	0.4	Carl Reed	All	Review

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Martin Daly: OGC 01-009, *Coordinate Transformation Services — OLE/COM*. Open Geospatial Consortium (2001). https://portal.ogc.org/files/?artifact_id=999.

[2] Akinori Asahara, Ryosuke Shibasaki, Nobuhiro Ishimaru, David Burggraf: OGC 14-084r2, *OGC® Moving Features Encoding Extension: Simple Comma Separated Values (CSV)*. Open Geospatial Consortium (2015). https://docs.ogc.org/is/14-084r2/14-084r2.html.

[3] Akinori Asahara, Ryosuke Shibasaki, Nobuhiro Ishimaru, David Burggraf: OGC 14-083r2, *OGC® Moving Features Encoding Part I: XML Core*. Open Geospatial Consortium (2015). https://docs.ogc.org/is/14-083r2/14-083r2.html.

[4] ISO: ISO 19148:2021, *Geographic information — Linear referencing*. International Organization for Standardization, Geneva (2021). https://www.iso.org/standard/75150.html.

[5] OGC: *OGC 22-035: Data Streaming Engineering Report*, 2022. http://ogc.pages.ogc.org/T18-3Dplus_Data_Standards_and_Streaming/documents/D026/22-035.html

[6] OGC: *OGC 22-036: 3D+ Standards Framework Engineering Report*, 2022. https://portal.ogc.org/files/?artifact_id=103466

[7] OGC: *OGC 22-038: Reference Frame Transformation Engineering Report*, 2022. https://portal.ogc.org/files/?artifact_id=103428

[8] IOGP: **EPSG Geodetic Parameter Dataset**. https://epsg.org

[9] NASA: *SPICE: Navigation and ancillary information facility*. https://naif.jpl.nasa.gov/naif/