

# **Open Geospatial Consortium**

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: <2022-02-11>

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

Internal reference number of this OGC® document: 22-000

Version: 0.9

Category: OGC® Engineering Guidance

Editors: C. Perey, J.G. Morley, J. Lieberman, R. Smith, M. Salazar, C. Smyth

## **OGC GeoPose Reviewers Guide**

### **Copyright notice**

Copyright © 2022 Open Geospatial Consortium

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

### **Warning**

This document provides guidance for reviewers of the OGC GeoPose Standard. This document is a non-normative resource and not an official position of the OGC membership. It is subject to change without notice and may not be referred to as an OGC Standard. In addition to this guide, developers, implementers and reviewers may wish to study the OGC GeoPose Users Guide. The guidance provided in this document is not to be referenced as required or mandatory technology in procurements.

Document type: OGC®Engineering Guidance

Document subtype:

Document stage: Draft

Document language: English

## License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

# Table of Contents

1. Introduction .....	5
1.1. What is GeoPose? .....	5
1.2. Why is Another Standard Needed? .....	5
1.3. How Does OGC GeoPose Address Diverse Requirements? .....	6
1.4. How Was the OGC GeoPose v 1.0 Scope Defined? .....	6
1.5. Who Will Use the OGC Reviewers Guide? .....	6
1.6. How To Use This Resource .....	7
2. Scope .....	8
3. Terms and Definitions .....	9
4. Conceptual Overview .....	10
4.1. Key Concepts .....	10
4.2. Orientation Systems .....	14
4.3. Positioning Systems .....	18
4.4. Time-related Concepts .....	21
5. The GeoPose Standard .....	26
5.1. Conceptual Model .....	26
5.2. Logical Model .....	26
5.3. Encodings for Implementation .....	27
6. OGC GeoPose in Context .....	30
6.1. OGC GeoPose Connects Objects to their Representation .....	30
6.2. GeoPose for 3D Asset Formats .....	31
6.2.1. The Khronos Group's glTF format and GeoPose .....	31
6.2.2. Pixar's open-source USD format and GeoPose .....	31
6.3. OGC GeoPose in the Landscape of Standards .....	32
6.4. Standards which Could Reference GeoPose in Normative Clauses .....	33
6.4.1. OGC Standards .....	33
6.4.2. Other SDOs .....	35
6.4.3. Space Science .....	38
7. Conclusion .....	40
Annex A: Revision History .....	41

## **i. Abstract**

The GeoPose Reviewers Guide is a public resource structured to provide quick answers to questions which a reviewer may have about the [OGC GeoPose specification](#). This OGC document is provided to support professionals who need to understand OGC GeoPose and/or are reviewing the GeoPose draft standard but do not wish to implement it.

[GeoPose 1.0](#) is an OGC Implementation Standard for exchanging the position and orientation of real or virtual geometric objects (Poses) within reference frames anchored to the earth's surface (Geo) or within other astronomical coordinate systems. The standard specifies two *Basic* forms with no configuration options for common use cases, an *Advanced* form with more flexibility for more complex applications, and five composite GeoPose structures that support time series plus chain and graph structures.

## **ii. Keywords**

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

GeoPose, ogcdoc, OGC document, OGC Implementation Standard, Geospatially-anchored position and orientation, pose, reviewers

## **iii. Preface**

This version of the GeoPose Reviewers Guide is limited in scope to the specification for GeoPose 1.0. Content of this document will be updated when relevant information and feedback to the OGC GeoPose 1.0 is provided and the standard updated. The Open Geospatial Consortium shall not be held responsible for the accuracy or completeness of this reviewers guide.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

## **iv. Submitting organizations**

The OGC GeoPose Standards Working Group submitted this document for publication by the Open Geospatial Consortium (OGC).

## **v. Submitters**

The OGC GeoPose Standards Working Group submitted this document for publication by the Open Geospatial Consortium (OGC).

# Chapter 1. Introduction

## 1.1. What is GeoPose?

Conceptually, when a real or digital object's pose is defined relative to a geographical frame of reference it will be called a "geographically-anchored pose." All physical world objects inherently have a geographically-anchored pose. Digital objects may be associated with a geographically-anchored pose (for example, in a real-world overlay or on a stage).

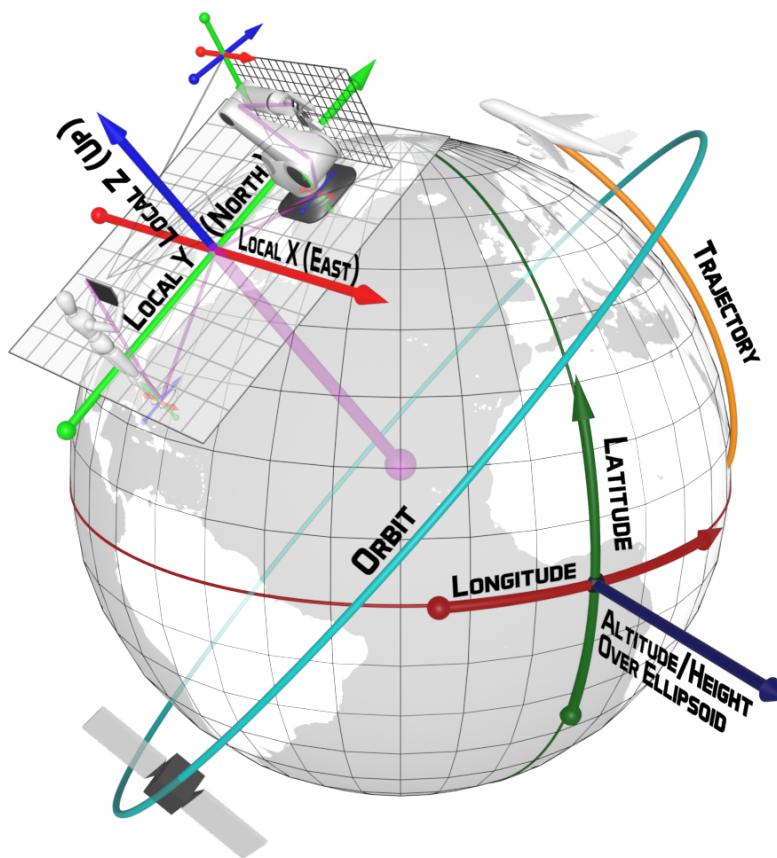


Figure 1. GeoPose Concepts

Specifically, the OGC GeoPose Standard defines the rules for the interoperable interchange of geographically-anchored poses. As such, the OGC GeoPose Standard defines a conceptual model, a logical model, and encodings for the position and orientation of a real or a digital object in machine-readable forms using real world coordinates.

## 1.2. Why is Another Standard Needed?

A new standard is required to facilitate the seamless interchange of position and orientation information between proprietary systems and any systems that implement to existing standards. The [review of standards related to GeoPose](#) demonstrates that there are many relevant specifications and standards that could use GeoPose, but do not, themselves, provide the elements and rules necessary to share poses.

## 1.3. How Does OGC GeoPose Address Diverse Requirements?

The [OGC GeoPose 1.0 standard](#) defines data structures and rules for the interoperable exchange of the position and orientation of real or virtual geometric objects (Poses) within reference frames anchored to the earth's surface (Geo). In developing this standard, the SWG sought to use a single conceptual model to address requirements ranging from common use cases that benefit from low complexity and low optionality ("without optional parameters"), to those complex use cases needing more flexibility and extensibility.

In order to meet the wide range of requirements, the OGC GeoPose Standard specifies:

1. Two basic forms with no configuration options for common use cases,
2. An advanced form with more flexibility for more complex applications, and
3. Five composite structures to support time series plus chain and graph structures.

OGC GeoPose 1.0 is the [OGC Implementation Standard](#) for exchanging GeoPoses.

## 1.4. How Was the OGC GeoPose v 1.0 Scope Defined?

While the earth is the focus of the GeoPose 1.0, the standard could also be used in conjunction with other astronomical bodies than the earth.

In the course of developing GeoPose v1.0 --and in order to focus on the key objectives of the standard - the SWG members decided that the following considerations would be out of scope for the v1.0:

- Details of any frame transformations (e.g., the radius of the Earth),
- Differential properties (i.e., acceleration and velocity) and other physical properties of objects that could be associated with a GeoPose,
- Concepts of uncertainty (accuracy and precision),
- Camera models or view frustums,
- Scaling and other non-rigid transforms,
- Interpolation methods in case of complex targets.

While not in scope for GeoPose 1.0, any of the above could be presented in parallel with GeoPose. For example, many of the aspects which are excluded could be introduced as more properties in a schema.

## 1.5. Who Will Use the OGC Reviewers Guide?

The GeoPose Reviewers Guide is a resource for those who seek to understand key OGC GeoPose concepts used in OGC GeoPose, the requirements that the standard meets and the data structures the standard specifies.

The OGC intends this guide to be useful for reviewers of the standard as well as decision makers seeking to understand the relevance of this standard in their use cases.

## 1.6. How To Use This Resource

The GeoPose Reviewers Guide is not intended to be read from start to finish. Rather, the document is a resource structured to provide quick answers to questions which a reviewer may have about the [OGC GeoPose specification](#). This guide is provided to support professionals who need to understand OGC GeoPose and/or are reviewing the GeoPose draft standard but do not wish or have need to implement the Standard.

In addition, this guide can provide insights to professionals considering adopting GeoPose for their projects and products.

The GeoPose Reviewers Guide contains hyperlinks which can be used to navigate directly to relevant sections of the guide as well as to sections of the GeoPose specification.

# Chapter 2. Scope

The GeoPose Reviewers Guide introduces the the [key concepts used in the GeoPose Standard](#) to its [target audiences](#).

To identify broadly applicable requirements for GeoPose, the SWG solicited use cases and chose five cases that were agreed to be representative. To understand the ways in which GeoPose can be used and how it meets requirements identified, this guide can be used in conjunction with the OGC GeoPose [use cases](#) section of the Standard.

The choices of [standardization targets](#) made in the GeoPose SWG during standard development of the Standard are explained in [this section](#) of the present guide.

Finally, this guide explains how GeoPose fits in the landscape of geospatial computing. The guide compares GeoPose with approaches that have been [taken in other standards](#) for encoding geospatially-anchored position and orientation with six degrees of freedom.

# Chapter 3. Terms and Definitions

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

For GeoPose, specifically, the reader will consult the [Conceptual Overview Section](#) of this Guide.

## 2D data

Geometry of features is represented in a two-dimensional Euclidian space

NOTE In other words, the geometry of 2D data is given using (X,Y) coordinates.

[INSPIRE D2.8.III.2, definition 1]

## 2D geographical data

Geometry of features is represented in a two-dimensional geographical

(i.e, ellipsoidal) space using (Longitude, Latitude) coordinates.

## 2.5D data

Geometry of features is represented in a three-dimensional space with the constraint that, for each (X,Y) position, there is only one Z

[INSPIRE D2.8.III.2, definition 2]

## 3D data

Geometry of features is represented in a three-dimensional space.

NOTE In other words, the geometry of 2D data is given using (X,Y,Z) coordinates without any constraints.

[INSPIRE D2.8.III.2, definition 3]

## conceptual model

Model that defines concepts of a universe of discourse

[ISO 19101-1:2014, 4.1.5]

## implementation standard

Specified on the [OGC Document Types Register](#).

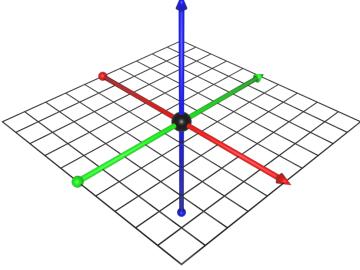
# Chapter 4. Conceptual Overview

This glossary was originally created to facilitate the communication between GeoPose SWG members and reach a common understanding of concepts related to GeoPose. It has been integrated into this guide to provide readers with a better understanding of the conceptual foundations on which the standard is based.

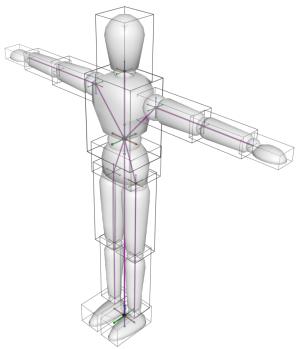
The definition of each term is accompanied by a visual representation of the concept, a more general explanation and examples of how it may be applied. Additionally, the text includes hyperlinks to connect the different concepts to one another and to illustrate how they are used in other standards and real world applications.

## 4.1. Key Concepts

The following list of terms covers the conceptual underpinnings for the GeoPose standard while offering examples of how they are actually employed in different scientific fields and industry sectors.

<p><b>(Reference) Frame</b></p>  <p><a href="#">Frame.blend</a></p>	<p><b>As defined in ISO 19111, a reference frame is a parameter or set of parameters that realize the position of the origin, the scale, and the orientation of a coordinate system.</b></p> <p>Another, more practical definition <a href="#">from Wikipedia</a> is: <i>"In physics, a (reference) frame consists of an abstract coordinate system and the set of physical reference points that uniquely fix (locate and orient) the coordinate system and standardize measurements within that frame."</i></p> <p>A reference frame can be visualized as a subspace with its own <a href="#">coordinate system</a> (represented with an arrow for each axis and a 2D grid to facilitate measurements) and a point of origin (represented as a sphere at the point where the arrows intersect). Generally, to simplify calculations, the associated coordinate systems are Cartesian in nature, but in the case of <a href="#">non-euclidean spaces</a>, they require more advanced definitions (like the <a href="#">geographical coordinate systems</a> used in global positioning systems).</p>
---	--

## Pose



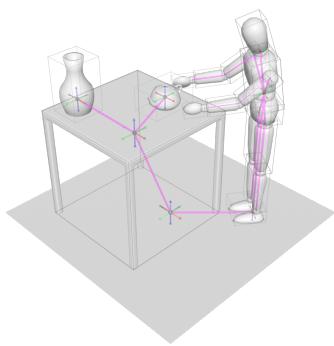
Pose.blend

"A pose determines the position and orientation of an object (or group of objects) relative to a coordinate system."

In robotics and human-computer interaction, this concept is often referred to as [Six Degrees of Freedom \(6DoF\)](#) and it is used to specify the position and orientation of an object with great precision (generally, by combining a **Local Positioning System** for the translation and an **Aircraft Principal Axes Orientation System** for the rotation). In this way, it is possible to define complex [kinematic systems](#) that allow the creation of advanced [robotic arms](#) and [exoskeletons](#).

From the point of view of computer graphics, however, a pose can be seen as a simplified version of a "transform class" (without the **scale**-related fields/properties). This direct correspondence implies that a pose is associated to a single node of a scene graph (i.e., the "hand" has a pose, the "arm" has another pose, etc.), but, when 3D artists and engineers talk about "poses", they are generally referring to the combined pose data of all the components of an entity (i.e., the pose of a dummy, as seen in the illustration). In fact, most 3D editing software solutions include a separate "pose mode" so that artists can independently work on the geometry of the different 3D models and on their animation.

## Pose Chain



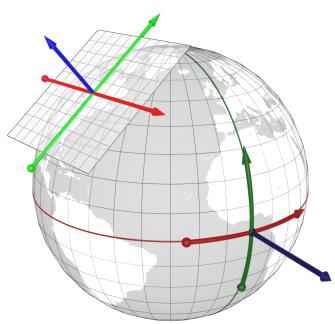
SceneGraph.blend

**"A pose chain is a sequence of poses that is used to determine the position and orientation of objects across multiple frames."**

When dealing with multiple reference [frames](#) (specially, when they are of different types), it is necessary to perform several complex transformation operations to convert between the different [Coordinate Reference Systems](#). To facilitate this time-consuming process and to ensure that the operations are applied in the right order, the different poses are often organized into one-dimensional sequences of [poses](#) called "chains".

In many applications, where a large number of entities have similar pose chains, it is common to reorganize this structure into a graph, so that it is possible to more effectively apply the transformation operations (by storing the result of the repeated operations). In computer graphics, this concept is the basis of the "scene graph", in which the transform class instances of the different entities are connected together using a hierarchical graph through (single) parent-children relationships. In this way, the pose of each entity of the graph is calculated in a [depth-first](#) manner that minimizes the number of required calculations.

## OGC GeoPose



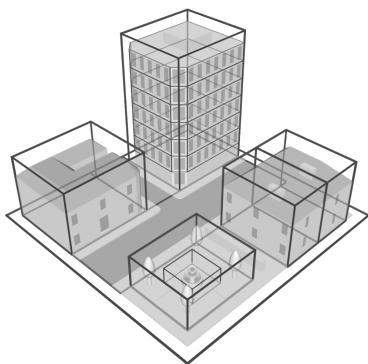
## GeoPose.blend

"A geographically-anchored pose is a [pose](#) defined in relation to a real-world object (by default, Earth), typically, via a geographical [Reference frame](#)."

When a "GeoPose" is provided, it will be understood to mean a geographically-anchored pose that complies with the OGC GeoPose specification. The goal of the standardization of OGC GeoPose is to establish a single mechanism to encode the position and orientation of any real/virtual objects within a geographical (ellipsoidal) [frame](#), greatly simplifying the determination of the pose of entities relative to an astronomical object and, consequently, relative to each other.

In practical terms, a GeoPose facilitates the application of one or more transformation operations to properly position and orientate modes in a [scene graph](#) (generally, the root one) on a realistic 3D map and/or in an Augmented/Mixed Reality environment.

## Location



location.blend

"Not to be confused with the term position, a location is a distinctive region of a space."

While a position merely specifies a single point within a space (relative to a [frame](#)), a location defines a limited region within a space (often with its very own [frame](#)). A location generally receives a unique name within a context (or within another location) to easily identify it from others and, if necessary, it can also have additional semantic information to further differentiate it (e.g., "Washington D.C.", "Washington State" and "Washington City, Utah"). There are as many types of locations as there are types of spaces (e.g., "the bottom of screen", "behind the car", "Earth's orbit", etc.), but it is specially relevant in the context of geography, where the \*geo\*location ("In which street/road/town I am?") is often much more important than the actual \*geo\*position ("what are my GPS coordinates?").

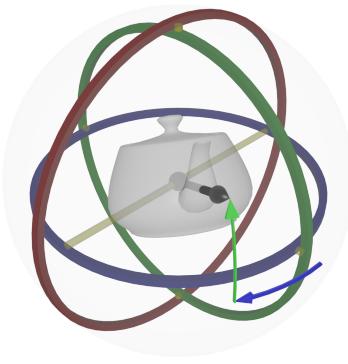
The boundaries of locations are often defined using either dimensional properties (i.e., width, height and depth) or specific shapes (most notably, 2D projections in a geographical space called [geofences](#)). However, when there are a large number of locations or these are constantly changing, the boundaries are defined by proximity to the closest point in the [topological skeleton](#) or by the minimum number of logical connections.

## 4.2. Orientation Systems

This section covers a system to accurately describe the orientation of an object relative to the associated reference frame. As with positioning counterparts, these systems provide a vector that needs to be consistently interpreted (so that it can be converted into a rotation matrix), not only to describe the numeric values in the proper sequence, but also to note the units and to specify the type of orientation system to which the values refer.

Not all applications require orientation information to define a pose.

## Euler Angles



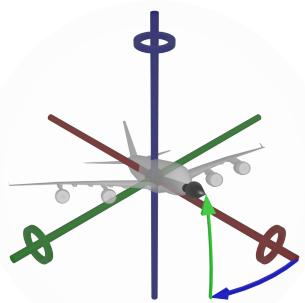
[EulerAngles.blend](#)

"The Euler angles are three angles introduced by Leonhard Euler to describe the orientation of a rigid body with respect to a fixed coordinate system." [\(from Wikipedia\)](#)

The angles are generally specified in degrees (in a similar way to longitude/latitude values in Global Positioning Systems), although the internal operations on computers are almost always performed in radians, one axis after another. Usually, the application order of rotation transformations is the inverse of the specification (the Z angle is applied first, then the Y one, and finally, the X one), but it might vary depending on the use case or the hand rule employed.

It is important to note that, since the rotation operations are applied globally and linked together by the application order, if two axes are driven into a parallel configuration, it can generate a [Gimbal Lock](#), resulting in the loss of one degree of freedom. This is usually not a problem when defining simple, discontinuous orientations, but when the use case involves overlapping or interpolating between multiple Euler angles, the result can be negatively affected.

## Aircraft Principal Axes



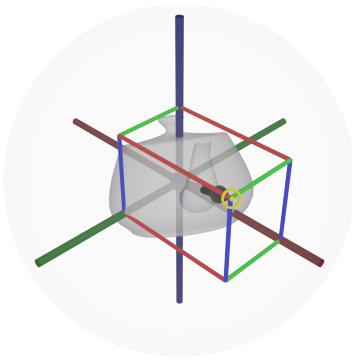
AircraftPrincipledAxes.blend

"Aircraft Principal Axes define the relative (local) rotation of an object, using a plane flying in the +X axis as a reference."

This system defines the orientation of an object as a combination of rotations along the three axes of an aircraft: yaw (or normal) axis, pitch (or transverse) axis and roll (or longitudinal) axis. These axes have a direct correspondence with the X, Y and Z axis of the (right-handed) Euler system and are defined with the same units. However, since these axes are applied locally (i.e., rotating the coordinate system of the object at each step), the order of application becomes irrelevant.

The Aircraft Principal Axes system has the advantage of being very intuitive (once the users fully understand what axis is related to the terms yaw, pitch and roll) and of not being susceptible to [Gimbal Lock](#) issues. On the other hand, this orientation system often requires additional computational power to generate the intermediate rotation matrices (one for each axis with a value different from 0) and, while the image of an airplane can be projected onto other vehicles, it might be difficult to do so with other static elements (e.g., it doesn't make sense to define the "roll angle" of buildings or any other static structures).

## Quaternions\*



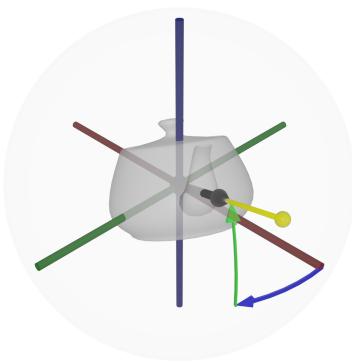
Quaternion.blend

"In computer graphics, a Quaternion is 4D complex number that is used to specify orientations in a 3D space."

In essence, a Quaternion is a mathematical construct with 3 imaginary components ( $i$ ,  $j$  and  $k$ ) and one real component. When it is used to describe 3D orientations, each imaginary component describes the dimension of a **unitary** vector in a particular axis of the space ( $i$ ,  $j$  and  $k$  are assigned to the X, Y and Z axis, respectively) whereas the real component expresses **half** of the rotation around that unitary vector, expressed in radians. However, to ensure that the resulting axial vector is unitary (i.e., its length equals one), a quaternion has to follow the fundamental formula  $i^2 = j^2 = k^2 = i \cdot j \cdot k = -1$ .

Due to their mathematical construction and the possibility to interpolate them both [linearly](#) and [spherically](#), most 3D engines use Quaternions internally to orientate and animate compound objects. However, the complex nature and the interdependence between imaginary components, make Quaternions very difficult for most human beings to understand and use, so they are rarely employed in user interfaces or human-readable documents.

## Look-At Systems



LookAtSystem.blend

"A Look-At system enables the definition of orientation relative to another object or position vector."

Similar to Quaternions, Look-At systems use a vector (obtained from the difference of the target position minus the current position) to determine the orientation of an object. After normalizing the vector, it is possible to use its components to create a quaternion and apply the same mathematical operations. However, to properly determine the rotation angle around the axial vector (the real component of a quaternion), it is necessary to also provide an "up" vector to serve as a pivot.

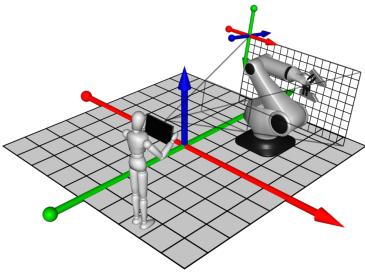
## 4.3. Positioning Systems

This section covers systems used to accurately describe the position of an object relative to the associated reference frame. The position of an object is a single point in space relative to the origin point of the reference frame. This point is usually placed in the center of mass of the object and provides a simple way to perform other spatial calculations (i.e., distances between objects, travel time, collision detection/avoidance, etc.). In contrast, with a camera, the point could be its focal point.

The objective of positioning systems is to provide a position vector (that is often converted into a translation matrix) as precise as possible within a reference frame. These vectors are often accompanied with an (in)accuracy value to be able to establish an uncertainty threshold for different tasks.

We distinguish the concepts of position and location. Location is a broader term that can be represented in many forms, including textual forms such as the address. While it is possible to define a position using a known location (e.g., "This building is in Tokyo"), this practice often results in imprecise information.

## Local Systems



LocalPositioning.blend

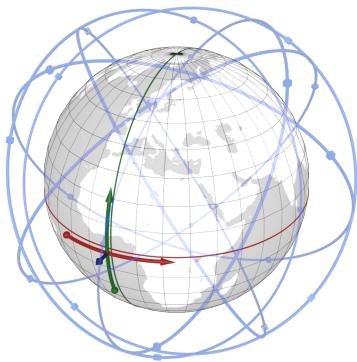
"A Local Positioning System defines the location of objects relative to a frame based on [Euclidean spaces](#)."

These frames are often defined with an arbitrary point of origin depending on the nature of the subspace to represent. When working with physical spaces or virtual 3D scenes, the point of origin is generally situated in the center of the space (to avoid the limitations of [single-precision floating-point numbers](#)).

However, when operating within 2D spaces with clear boundaries, like screens or documents, the point of origin is normally placed in the top-left corner (due to the graphic systems based on the left-to-right, top-to-bottom writing direction used in most western countries). Needless to say, this makes the transformations between the different frames (i.e., determining what 3D object has the user selected on a 2D screen) a very cumbersome process.

In recent years, advancements in [Indoor Positioning Systems](#) technologies have enabled the precise tracking of real-world elements in relatively large **local** subspaces. Yet, when a use case has more than a single location or the curvature of the Earth becomes an important factor to take into account, using a Global Location System is generally a better option.

## Global Systems



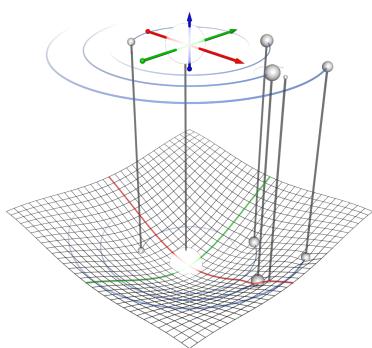
GlobalPositioning.blend

"A Global Positioning System defines the location of objects relative to a frame based on **spherical spaces**, generally associated with the planet Earth."

As its name implies, a Global Positioning System is defined by a spherical object that establishes the properties of a **frame** itself (even if the astronomical body is not a perfect sphere). The resulting subspace is delimited in the two horizontal dimensions of the surface of a sphere (**longitude** should only have values between -180 and 180 degrees, whereas the **latitude** values should be within a -90 to 90 degrees) and partially delimited in the vertical dimension (**altitude** values should never be lower than the negative value of the radius of the sphere). As for the point of origin of this subspace, for practical reasons, it is generally placed on an arbitrary point alongside the equator of the sphere (thus, also defining an offset value for the altitude values).

Traditionally, determining the position of an object on the surface of the Earth with any degree of accuracy required cumbersome tools and manual calculations. Fortunately, satellite-based radio-navigation systems like [GPS](#), [Galileo](#) or [NavIC](#) have greatly simplified the process and -as long as the devices can establish visual contact with three satellites- it is possible to track the position of all kinds of objects on Earth.

## Universal Systems



UniversalPositioning.blend

"A Universal Positioning System defines the location of objects relative to an inertial frame based on [Minkowski spaces](#)."

These systems aim to provide a more precise and truly unique positioning for objects relative to an inertial point in the physical Universe (generally, the barycenter of the Solar System). However, since the Minkowski spaces are defined by the rules of [Special Relativity](#), the very subspaces are "curved" by [Lorentz Transformations](#) and any traversal is limited by the [Speed of Light](#).

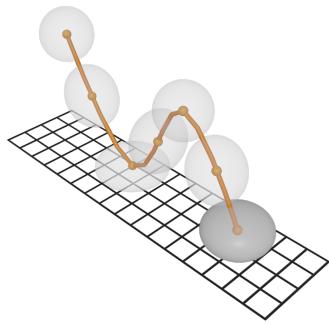
Needless to say, the extremely complex calculations that these positioning systems require relegate them to applications related with interplanetary travel and communications. For example, the SPICE framework created by the NASA's Planetary Science Division operates with both the [J2000](#) and [ICRF Inertial Frames](#) to calculate transfer orbits between bodies of the Solar System.

## 4.4. Time-related Concepts

While, by definition, a pose defines the position and orientation of an object at a particular point in time, there are many applications that require the use of temporal sequences of poses. Properly taking into account the temporal dimension requires consideration of additional factors like (moment of) inertia, mass, changes in shape and, in extreme cases, even the variations in the Earth's gravity field.

To help represent this complexity, the following list of terms is intended to offer a basic introduction to the topic.

## Animation



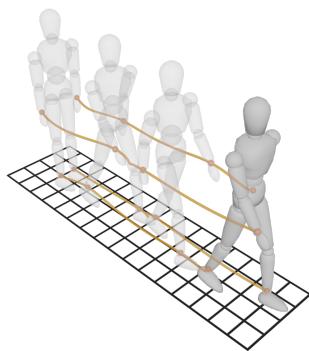
[Animation.blend](#)

**"Animation is a method to create the illusion of movement through a collection of still images."**

In computer graphics, animation is generally simulated by modifying the properties of an object (including its position, rotation, scale, material colors, etc.) over a period of time (or timeline). Since complex deformations require a large number of physics calculations, most animation systems either operate with properties independently (i.e, separating position, orientation and scale) or define skeleton-based systems to alter the geometry of the object in a controllable way. In recent years, however, many 3D engines incorporate procedural subsystems to simulate advanced animations (such as explosions, wind systems, collision deformations, etc) in real-time, with relatively high degree of precision.

Internally, each change in properties is stored into an "animation key," and when there is more than one key over time, an "animation curve" is generated to specify how to interpolate between them. A collection of curves is called an "animation clip" and can be associated to a given action or event.

## Motion



## Motion.blend

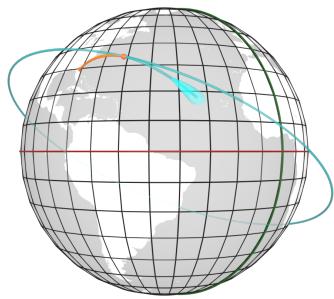
**"A motion is a type of animation that only defines the movement (position+orientation keys) of objects."**

In many use cases, it is possible to simplify complex animations by segmenting an object (generally, using a skeleton structure) and reducing the types of keys to just position and orientation (creating animation curves known as Motion Paths). This idea of only taking into consideration the movement of different parts of an object is the basis of [motion capture](#) techniques; a series of technologies that can even [record subtle facial expressions of human beings](#).

In addition, motions can also be defined as transitions between different poses of an object. In this way, instead of having to define a single sequence of animation keys, it is possible to "jump" between different poses by merely interpolating between them whenever necessary (something especially useful when dealing with real-time camera -motion- tracking systems).

It is important to note that a motion between two poses also generates a direction vector and an attitude (orientation relative to that direction vector).

## Trajectory



### Trajectory.blend

"A trajectory is a precalculated motion path that attempts to describe the future movement of an object."

While motion paths describe the actual movements over time, in many use cases it is imperative to also be able to predict the future movement of an object (e.g., to avoid collisions or reach a location/orbit with a very specific speed and orientation). These special motion paths are called **trajectories** and are usually calculated by simply taking into consideration the inertia of an object.

This is especially important for objects placed in stable trajectories relative to other objects (usually, astronomical bodies), because if those trajectories are entirely predictable, it is possible to create an [Ephemeris](#) system to rapidly calculate their position at any point in time. Additionally, if a predictable trajectory is regularly repeated over time, it can also be called a stable [orbit](#).

## Gesture



Gesture.blend

**"A gesture is a pose or motion used to convey a non-vocal message."**

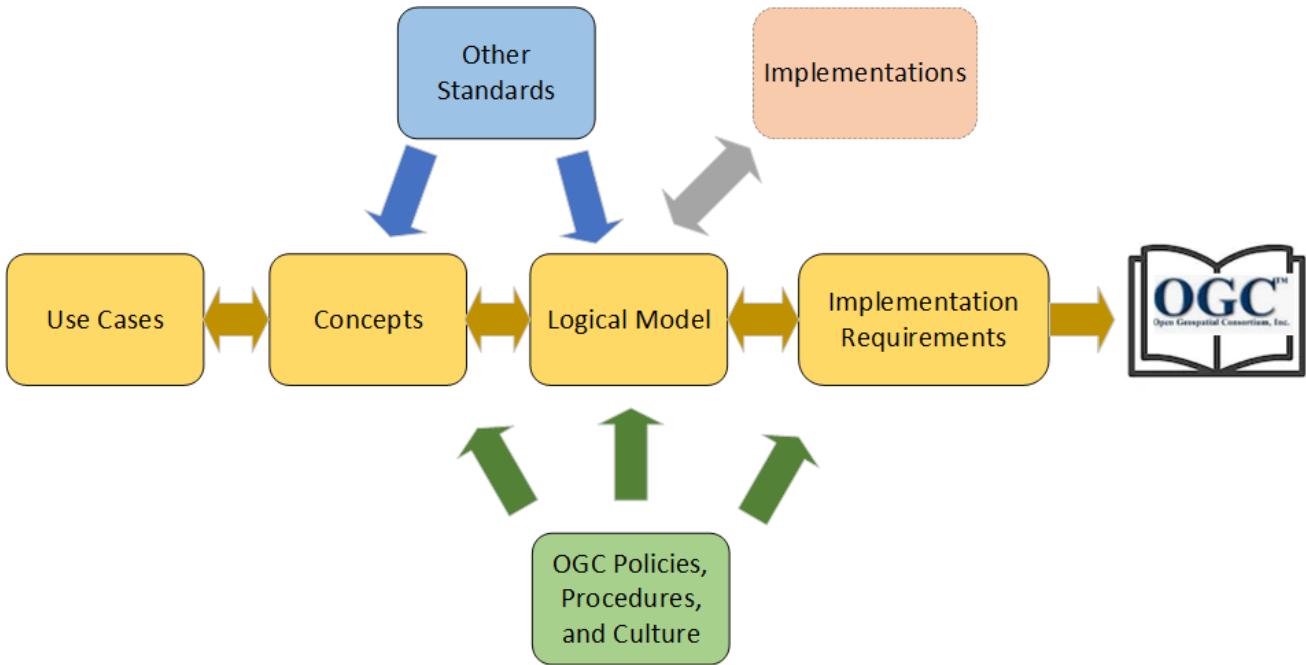
Gestures are a form of communication that involve the positioning/movement of a part of the body in a predetermined way to express a particular concept. And although this type of communication is used by many living beings, humans have used them to develop complete sign languages or to serve as a method of identification. Some basic gestures even allow people to overcome linguistic barriers and create a culture around specific gestures.

Nevertheless, since messages conveyed by gestures require both the issuer to be able to make the gestures and the receiver to be able to interpret them correctly, both the communication channel (the physical space) and the meaning of each gesture has to be previously defined to avoid misunderstandings.

It is important to clarify that gestures can also be conveyed by a visual representations (e.g., the Unicode symbol U+1F44C "👉") or with robotic systems that emulate the body structure and movements of the living beings (which is one of the basis of the animal-robot interaction field).

# Chapter 5. The GeoPose Standard

This section describes the key elements of the GeoPose standard, especially the conceptual and logical models, and the implementation targets which have been derived from the logical model. The development of the standard has been led by a number of use cases and use case domains, as summarised in the diagram below.



## 5.1. Conceptual Model

The GeoPose Conceptual Model consists of linked definitions of terms denoting concepts expressed in the GeoPose Logical Model and structural data unit specifications for the implementation targets.

The Conceptual Model is defined in [this section of the GeoPose Standard](#).

## 5.2. Logical Model

While the Conceptual Model outlines the component terms and relationships in the standard, the Logical Model precisely models the classes and their relationships.

The Conceptual Model describes a (non-normative) domain of discourse for terms used in defining a precise Logical Model (normative) expressed as a Unified Modeling Language (UML) [ref] class diagram.

The scope of the Logical Model is a subset of the scope of the Conceptual Model. The scope of the implementation targets is a subset of the scope of the Logical Model.

The Implementation Targets are mutually independent implementations of subsets of the Logical Model.

## 5.3. Encodings for Implementation

The core of the OGC GeoPose Standard is the abstract frame transform. This is a representation of the transformation taking an outer frame coordinate system to an inner frame coordinate system. This abstraction is constrained in GeoPose v 1.0 to only allow transformations involving translation and rotation. The intention is to match the usual concept of a pose as a position and orientation. The formalism that expresses a GeoPose frame transform is a pair of reference frames, outer and inner, each defined by a frame specification.

Conformant implementation of the Standard (position and orientation) is accomplished by implementing one or more of the eight conformance classes. The requirements for each conformance class are documented in eight requirements classes. Each requirements class has a one-to-one correspondence with one of the eight standardization targets defined in the Standard.

*Summary of the Eight OGC GeoPose Standardization Targets:*

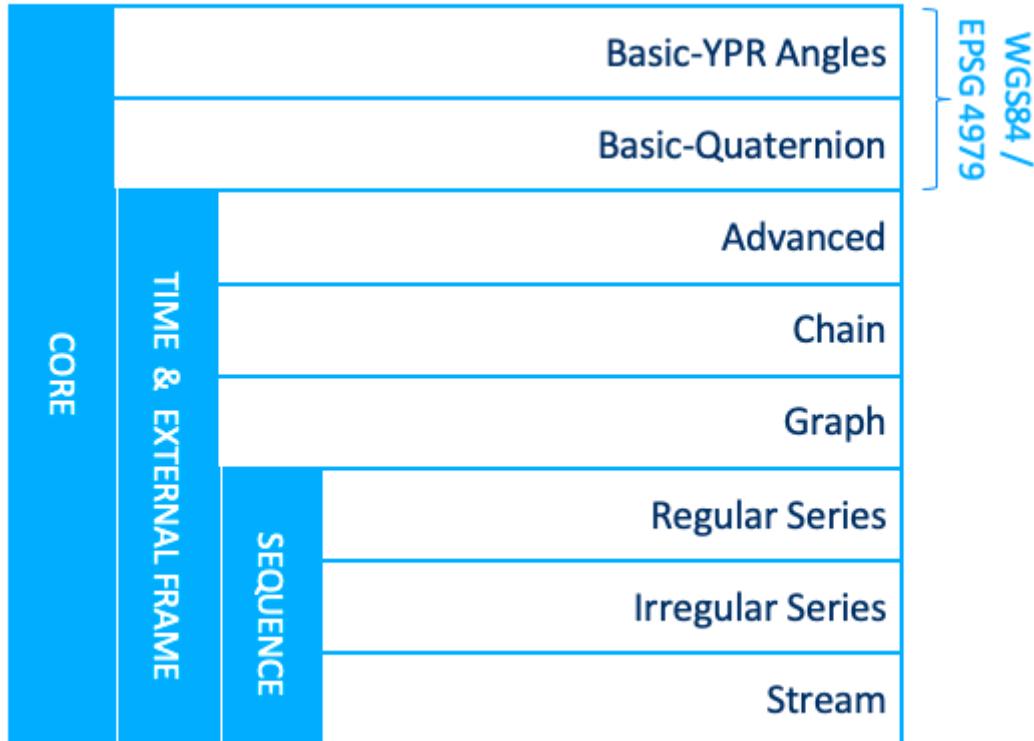
Core:

- The two Basic GeoPose targets are simple and concise – no options. They satisfy most use case requirements and assume a local tangent plane ENU frame derived from WGS84.
- An Advanced GeoPose target supports more complex use cases where the outer geographic reference frame is not LTP-ENU and/or a valid time is needed.

Composite:

- The Chain GeoPose target provides additional flexibility with multiple intermediate frames or specific coordinate reference systems as needed.
- The Frame Graph target supports the full structure need to represent networks of reference frames that arise with the use of multiple and linked location technologies.
- The three (Time) Sequence targets support the packaging of fixed-length time series of GeoPoses and the payload data objects for open-ended GeoPose streams.

In the figure below, the dependencies between targets are illustrated.



### Logical Elements

Figure 2. GeoPose Target Dependencies

The target names and short descriptions appear in the table below.

Core	Simple and concise, Basic targets cover most use case requirements and assume a local tangent plane east north up (LTP-ENU) frame derived from WGS84 (EPSG 4979)
<b>Basic-YPR Angles</b>	Inner Frame oriented by Yaw, Pitch and Roll angles using the aircraft principal axes-based orientation (i.e., consecutive rotations around the z, y and x axes, in that order)
<b>Basic-Quaternion</b>	Inner Frame oriented by unit quaternion
<b>Advanced</b>	Supports use cases where the outer geographic reference frame is not LTP-ENU/WGS84 and/or a valid time is needed
Composite	Where multiple, linked coordinate reference frames or sequences of data are to be represented
<b>Chain</b>	Provides additional flexibility with multiple intermediate frames or specific coordinate reference systems as needed
<b>Frame Graph</b>	Support the full structure need to represent networks of reference frames that arise with the use of multiple and linked location technologies
<b>(Time) Sequence</b>	Support the packaging of fixed-length time series of <u>GeoPoses</u> and the payload data objects for open-ended <u>GeoPose</u> streams <ul style="list-style-type: none"> <li>• Regular Series</li> <li>• Irregular Series</li> <li>• Stream</li> </ul>

Figure 3. Table of Eight OGC GeoPose Implementation Targets

Depending on the use case and requirements, a developer can implement support for one or more of these targets.

There are automated tests that can be used to determine whether an actual JSON-encoded GeoPose data object conforms to the Standard.

In the majority of use cases, requirements are met by using either:

- **Basic-YPR** GeoPose encoding when orientation is defined by yaw, pitch, and roll angles
- **Basic-Quaternion** GeoPose encoding when orientation is defined by a unit quaternion

There are two versions of Basic GeoPose since some developers prefer a human-readable encoding. In these cases, the Basic-YPR is chosen, in which the orientations are stored as angles. In general, orientation as quaternions in the Basic-Quaternion version is preferred and leads to less complex code.

The two differences between Basic GeoPose and Advanced GeoPose is that, first, in Advanced GeoPose, the developer can specify an outer geographic reference frame other than LTP-ENU. In addition, in the Advanced GeoPose encoding, a single time stamp can be provided. The time stamp pertains only to the single encoding so one of the Composite GeoPose encodings is not needed.

The three Composite GeoPose encodings are designed for when there are linked and sequential GeoPoses. The Chain GeoPose is a linear set of linked poses. The Graph GeoPose is used when there are linked poses but not necessarily in a linear sequence.

For time series poses with constant time spacing, the developer will choose to use the Regular Timeseries GeoPose encoding. When there is a per-GeoPose time stamp that is not at a regular interval, the developer will choose to use the Irregular Timeseries GeoPose encoding. Finally, in use cases that do not have a pre-defined end time (also referred to as an open-ended sequence of time-stamped GeoPoses), the developer will specify the Stream GeoPose encoding.

# Chapter 6. OGC GeoPose in Context

In this section of the reviewer's guide, the Standards Development Organizations (SDOs) and standards that specify data objects that convey the same information as one or more of the eight forms of GeoPose are listed. As a convention for the remainder of this section, the term "GeoPose" is used as a shorthand for any of the specified forms of the OGC GeoPose Standard.

The key question this section answers is where and how GeoPose fits into the current landscape of standards. By examining some of the relevant standards, this section illustrates the gaps which GeoPose fills.

There are many existing and emerging standards for position and orientation information. They have emerged from requirements defined in different industries: Aviation, planetary sciences, maritime, robotics, autonomous driving, satellite positioning, aerospace, and so forth. There is good practice in commercial and other domains for expressing the position and orientation of entities in all these fields. These existing standards cover different scales, for different purposes, different information environment - sometimes graphical, sometimes geospatial.

## 6.1. OGC GeoPose Connects Objects to their Representation

There are both conceptual and actual data pipelines that connect the real world and the objects in it with the representations in information and graphics systems. A number of the critical standards define how those pipelines are developed and interoperate or fit together.

In a standard Web browser, information is displayed on a plane. There is a need for a standard to represent information retrieved from the Web on the 2D display plane in a manner that is sufficiently fast to provide XR experiences. [W3C WebXR](#) focuses on this need.

In order to establish the data flow required for 2D to XR experiences, there is a need in graphics systems to locate those objects within a 3D local stage (or scene) so that when the user's head or perceived objects move within that "stage", the graphics systems can address those movements and changes correctly. [Khronos OpenXR](#) focuses on that stage of the pipeline.

In parallel, the OGC has been working on how sensor systems measure these changes in position and orientation. How does a system model the sensor that is capturing all the data about the features (e.g. objects) in its environment, including where they are and how the system represents them in a local 2D or 3D environment? OGC Sensor Web Enablement, particularly the OGC Abstract Specification Topic 20: Observations Measurements and Samples and SensorML Standards, addresses these needs.

Each of these stages in the pipeline needs to exchange data between themselves. How do you get the position and orientation of anything from anywhere to anywhere, into and out of these interactive environments? That's where GeoPose comes in. GeoPose defines the data structure(s) to pass position and orientation information between elements in the pipeline.

## 6.2. GeoPose for 3D Asset Formats

The world of 3D asset formats is highly complex. New formats are continuously being developed as new modeling and content creation applications and 3D content distribution platforms appear. This problem increases costs in 3D asset pipelines as well as leaving lots of stakeholders frustrated. The need for standardization is apparent.

Today, two complementary 3D asset formats are becoming widely adopted. The Khronos Group's glTF open standard is a transmission format to enable the efficient and reliable distribution of 3D assets to a wide variety of platforms including both Web and native applications. The open source USD format published by Pixar is an authoring format that enables teams of designers to collaborate on the creation of 3D assets.

In the spatial computing industry and, in particular, for use cases handling 3D assets in real-world anchored scenes, there is not (to the knowledge of this working group) a standardized solution for encoding a real world pose of 3D assets. Stakeholders who either work with or develop proprietary 3D formats or standards for 3D asset formats, should consider creating extensions that comply with the OGC GeoPose Standard to their formats.

### 6.2.1. The Khronos Group's glTF format and GeoPose

The Khronos Group is an important standards organization developing 3D and XR run-time APIs and 3D assets formats, including OpenGL, WebGL, Vulkan, OpenXR and glTF. glTF 2.0 enjoys support by a wide community and growing number of tools, engines and applications and has been submitted to ISO/IEC JTC 1 to become an international standard.

An extension to include GeoPose in glTF would be an important step for interoperability between the 3D computer graphics and geospatial industries.

### 6.2.2. Pixar's open-source USD format and GeoPose

Pixar, the company responsible for a long list of blockbuster 3D animated movies from the 1995 "Toy Story" to the upcoming feature film "Turning Red," created the whole genre of full-length 3D animated movies while creating an impressive number of technology breakthroughs in the fields of 3D graphics rendering and animation. Pixar engineers then unified and standardized the way different systems used by different people inside their organization could work on the same highly complex scene without the hassle of converting to and from different file formats and importing and exporting them between different systems. Pixar developers came up with the [Universal Scene Description](#) format (abbreviated USD). Realizing in 2016, that the solution to their internal interoperability issues would potentially solve industry-wide interoperability problems of 3D graphics and animation, they [open sourced](#) their code and documentation for the format. In 2018, Apple decided to use a zip-archive version of USD for 3D assets used in AR applications running on the iOS platforms. In 2019, NVIDIA announced they would standardize on USD for their Omniverse platform. Omniverse is being adopted for a range of industries such as autonomous vehicles, robotics, digital twins and city planning. In 2021, support for USD export and import was added to a range of 3D authoring software applications such as 3DS Max, Maya and Blender.

Although USD is not specified by a traditional standards body, it is developing into a de-facto authoring standard with wide adoption across a wide range of industries, and the wider tech world

can contribute through the [USD repo on GitHub](#).

Just like with the glTF format, USD allows for the creation of extensions. One potential avenue for the GeoPose SWG to explore would be to propose a collaboration with NVIDIA. NVIDIA has already gained experience creating extensions for USD and has a lot of industry customers who work in the geospatial domain who would be likely to find such an extension useful.

## 6.3. OGC GeoPose in the Landscape of Standards

The primary source of inspiration for GeoPose was the NASA SPICE framework. This framework is able to cover any scale from interplanetary and interstellar to specific local objects. NASA designed SPICE to address significant challenges in looking at both ephemeris objects (fixed, or on predictable paths) and objects that have changeable positions and orientations such as satellites, cameras, and other sensors. Representing different frames for these objects and being able to transform between them is really useful. SPICE is a formalism that is much larger than required for a simple or basic implementation, but is an incredibly appropriate foundation from which GeoPose evolved.

When GeoPose was designed, the OGC Moving Features Standard was considered. Although Moving Features describes object position and orientation, that standard is focused on a particular set of use cases with an emphasis on sensor streams, digital exhausts, and location information (usually GPS) coming off of vehicles in a municipal environment. Moving Features accomplishes this compactly so that the data can be easily incorporated into analytical and visual applications.

OGC Moving Features and GeoPose have distinct roles and are complementary. Moving Features is focused on a local, municipal scale, and rapid streams of measurements. Getting observations in and out of municipal management and other platforms easily and efficiently is one of the roles that GeoPose plays.

The OGC Sensor Web Enablement (SWE) suite of standards deals with how to work with sensors and requesting observations in a standard encoding.

- Concerns itself with sensors that measure observed properties of features of interest
- Defines models and encodings for elements of observations and measurements.
- Emphasis is on observing properties (position in Sensor Model is secondary concern)

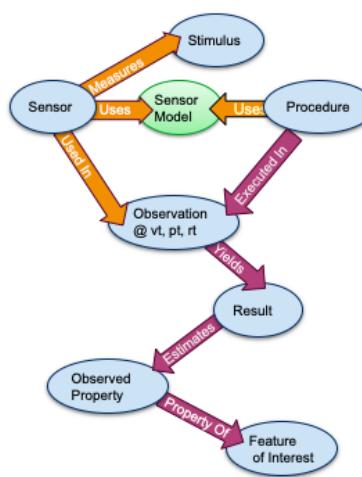
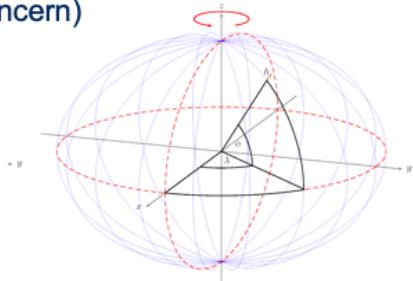


Figure 4. OGC GeoPose and Sensor Web Enablement

On the right of the above figure is a schematic showing the workflows that are enabled in Observations and Measurements, in Semantic Sensor Networks, and with encodings such as SWE

Common that enable transport of sensor outputs (observations). Another important part of these standards, SensorML, models the sensor process itself, from an initial environmental stimulus to how a measurement is recorded as an observation and recognized as an observed property of a real world object.

If sampling from a directed sensor (e.g. camera), orientation is an essential aspect of the sensor model. The result potentially includes the positions and orientations of both the sensor / platform and any observed entities in the world. However, the sensor position and orientation may be secondary to the primary objective of observing the positions and orientations of these observed entities, such as cars. GeoPose permits systems to get the position and orientation in and out of SWE-compliant platforms or devices without losing any resolution or introducing delays.

Pose is essential for combining physical and digital entities in visual scenes for them to be used by a service or a user, particularly if entities are being brought into a scene from very different source contexts, and regardless of where they are in space.

Khronos OpenXR also handles poses, particularly within specific spaces (frames of reference). It defines a set of reference spaces (view, local and stage); and specifies the model in which graphics hardware can use the pose in rendering objects. Within a particular graphical system, it is effective but GeoPose adds the capability to bring in a pose from any source in any frame of reference.

GeoPose can also relate the frame of reference of a Web browser window to a virtual world or the real world.

Geocentric (earth-based) position and orientation are the basis for all these integrations. OGC GeoPose provides that usable common ground, both the geospatial expertise that OGC has cultivated for many years and digital representation of physical space as the most common denominator among all these systems and representations.

To summarize, there are a number of well-developed standards for position and orientation. What these lack is a means for position and orientation information to be passed between them in a manner that is independent of graphical system, applications scene, frame of reference, and technology. OGC GeoPose offers portability of information between all these domains and systems.

The approaches to this issue that have been published in other standards prior to introduction of GeoPose appear in the tables below.

## 6.4. Standards which Could Reference GeoPose in Normative Clauses

### 6.4.1. OGC Standards

The OGC has many standards that express requirements for positioning and location. Some also express orientation. They do so in different scales and with different global and/or local coordinate reference systems. Some also deal with different time scales. However, these standards are not designed for sharing both position and orientation.

Some OGC standards (such as CDB) deal with fixed infrastructure. Other OGC standards, such as KML and IndoorGML, deal with somewhat more specialized information. Yet other OGC standards

deal with expressing location and orientation in very dynamic and real time scales, such as Sensor Web Enablement and Moving Features.

The GeoPose SWG's assessment of the pose-related elements of the OGC standards is summarised in the table below with the following headings:

- **Graphic/Virtual Context:** Does the standard relate only to a computer graphics context, abstracted from the outside world, or does the standard deal with virtual models of the real world?
- **Local SRS:** Does the standard allow a local spatial reference system (SRS), independent of a wider geographical framework, to be used to define coordinates?
- **Geodetic CRS:** Does the standard allow a spatial reference system connected to the shape of the Earth or its gravity field as the basis for coordinates?
- **6DOF as entity or attributes:** If the standard stores position and orientation for objects, is this information stored as attributes of the objects or does it use a pose entity in the data model with an association between the pose entity and the object entity?
- **Temporality:** Does the standard manage temporal information for the objects represented?

Standard	Graphic/Virtual Context	Local SRS	Geodetic CRS	6DOF as entity or attributes?	Temporality	Remark
Moving Features	Virtual	Y	Y	Attributes of temporal geometry	Y	
Sensor Web Enablement (SWE)	Virtual	Y	Y	Attributes	Y	
GML	Virtual	Y	Y	Attributes. Direction only (no roll)	Y	v3.2.1
CityGML	Both	Y	Y	Attributes	Y	
IndoorGML	Virtual	Y	Y	No orientation	IndoorPOI extension	Based on GML
CDB	Both	Y	Y	Y	Y	
KML	Both	Y	Y	Entity	Y	
Observations, Measurements and Samples (OMS)	Both	Y	Y	Both	Y	
SensorThings API	Virtual	Y	Y	Attributes	Y	
Indoor Mapping Data Format (IMDF)	Both	Y	Y	Attributes	Y	
3D Tiles	Both	Y	Y	Entity	Y	Based on glTF

Standard	Graphic/Virtual Context	Local SRS	Geodetic CRS	6DOF as entity or attributes?	Temporality	Remark
JSON-FG	Virtual	Y	Y	Attributes (Geometry, Where)	Y	Under Development

Note: 3D Tiles is basically a binary, encapsulated glTF with georeferencing.

Note: In OMS, where location is foreseen, it is always an ISO19107 Geometry Type. Temporal information is provided both in the Observation and the Deployment, utilizing TM\_Object or TM\_Period. Temporal attributes are defined in ISO19108.

#### 6.4.2. Other SDOs

There are other standards development organizations (SDO's) that deal with location and orientation for graphics. Information on these SDOs is compiled in the tables below. Work done in the W3C defines how systems express location and orientation for browsers. The Motion Imagery Standards Board (MISB) has standards for moving cameras. ISO also has sections of its standards in SC 24, such as the X3D standards, that encode orientation and position in graphics. In the Khronos Group, there are standards such as OpenXR and glTF that specify how to form digital assets that encode position and orientation

*Khronos Group*

Standard	Graphic/Virtual Context	Local SRS	Geodetic CRS	6DOF as entity or attribute?	Temporality
glTF	Both	Y	N	Entity	Y
OpenXR	Virtual	Y	N	Entity	Y

This [OpenXR Extension for Microsoft Spatial Anchors](#) allows an application to create a spatial anchor, an arbitrary freespace point in the user's physical environment that will then be tracked by the runtime. The runtime should then adjust the position and orientation of that anchor's origin over time as needed, independently of all other spaces and anchors, to ensure that it maintains its original mapping to the real world.

Standard	Graphic/Virtual Context	Local SRS	Geodetic CRS	6DOF as entity or attribute?	Temporality	Remark
Geolocation API	Virtual	No	Position & heading only	Attributes	Yes	Working Draft (Nov 21)
Orientation Sensor	Virtual	Orientation only	Orientation only	Attributes	Yes	Editor's Draft (Nov 21)
WebXR Device API	Virtual	Yes	No	Entity	Yes	Working Draft (Nov 21)

Levels of support for HTML features in current web browsers can be gauged from [W3C Web Platform Test results](#).

From the Immersive Web WebXR Device API documentation: [XRSpace](#) and [XR Pose](#) An XRSpace represents a virtual coordinate system with an origin that corresponds to a physical location. Spatial data that is requested from the API or given to the API is always expressed in relation to a specific XRSpace at the time of a specific XRFrame. Numeric values such as pose positions are coordinates in that space relative to its origin. The interface is intentionally opaque.

### Web 3D Consortium

Standard	Graphic/Virtual Context	Local SRS	Geodetic CRS	6DOF as entity or attribute?	Temporality
X3D Geo Component	Both	Y	Y	Attributes	Y

### Motion Imagery Standards Board (MISB)

Standard	Graphic/Virtual Context	Local SRS	Geodetic CRS	6DOF as entity or attribute?	Temporality
MISB ST 0601	Virtual	Sensor position & orientation relative to platform	Platform position & orientation	Attributes	UTC & media time
MISB ST 0801	Virtual	No	Camera position & orientation	Attributes	UTC & media time

*Third Generation Partnership Project (3GPP)*

<b>Standard</b>	<b>Graphic/Virtual Context</b>	<b>Local SRS</b>	<b>Geodetic CRS</b>	<b>6DOF as entity or attribute?</b>	<b>Temporality</b>
3GP (26.244)	Virtual	No	Y	Attributes	Media time

*Camera & Imaging Products Association (CIPA/JEITA)*

<b>Standard</b>	<b>Graphic/Virtual Context</b>	<b>Local SRS</b>	<b>Geodetic CRS</b>	<b>6DOF as entity or attribute?</b>	<b>Temporality</b>
Exif	Virtual	No	Position & heading only	Attributes	UTC

*International Organization for Standardization (ISO)*

<b>Standard</b>	<b>Graphic/Virtual Context</b>	<b>Local SRS</b>	<b>Geodetic CRS</b>	<b>6DOF as entity or attribute?</b>	<b>Temporality</b>
Spatial Reference Model (18026)	Virtual	No	Position only	Attributes	UTC
MPEG Immersive Video (MIV) MPEG-I (23090)	Virtual	No	Yes	Attributes	Media time
ISO19107	Virtual	Y	Y	3DOF	N
ISO19108	Virtual	Y	Y	N	Y

*Institute of Electrical and Electronics Engineers (IEEE)*

<b>Standard</b>	<b>Graphic/Virtual Context</b>	<b>Local SRS</b>	<b>Geodetic CRS</b>	<b>6DOF as entity or attribute?</b>	<b>Temporality</b>	<b>Remark</b>
Distributed Interactive Simulation (1278)	Virtual	No	Position only	Attribute	Y	
Smart Transducer Interface (1451)	N	Y	Y	Entity	Y	Based on GML

*BuildingSmart*

<b>Standard</b>	<b>Graphic/Virtual Context</b>	<b>Local SRS</b>	<b>Geodetic CRS</b>	<b>6DOF as entity or attribute?</b>	<b>Temporality</b>
Industry Foundation Classes (IFC)	Y	Y	Y	No	No

IfcSite and other IfcProducts permits topologic orientation, but not 6DOF. IFCSite lets users provide the WGS84 location (lat,lng,alt) of "the single geographic reference point for ([this site](#))"

For orientation they refer to the concept of "true north": "The world coordinate system, established at the IfcProject.RepresentationContexts, may include a definition of the true north within the XY plane of the world coordinate system, if provided, it can be obtained at IfcGeometricRepresentationContext.TrueNorth."

*ASTM*

Standard	Graphic/Virtual Context	Local SRS	Geodetic CRS	6DOF as entity or attribute?	Temporality
E57 3D Imaging Data Exchange (E2807-11 (2019))	No	Y	Y	Attribute	Y

The core capabilities of the E57 data exchange format [includes fifteen features](#).

#### 6.4.3. Space Science

There are also specifications (standards) that are developed for and used by industries/domains.

The Observation Geometry System used by NASA for space science missions is called SPICE. The OGC GeoPose SWG has chosen to make reference particularly to SPICE in this document. SPICE is an elegant system which ties together ephemeris information (including position and orientation data) in contexts ranging from the Earth system through to spacecraft, solar system (with the '[J2000](#)' fundamental inertial reference system) and planetary bodies. The SWG members were inspired by some of the concepts, particularly the ideas of frame of reference transformations and of satellites, constellations of satellites and other objects in orbit around the Earth. SPICE handles complex situations such as the relative pointing of spacecraft in motion around other bodies in the Solar System - this flexibility points to a complete and elegant solution.

While the SWG appreciated the full treatment of frame transformation in the SPICE system, the OGC SWG members took the approach of reducing the scope to Earth-based systems in version 1 of the GeoPose standard. The intention has been to permit later extension to a wider solar system viewpoint.

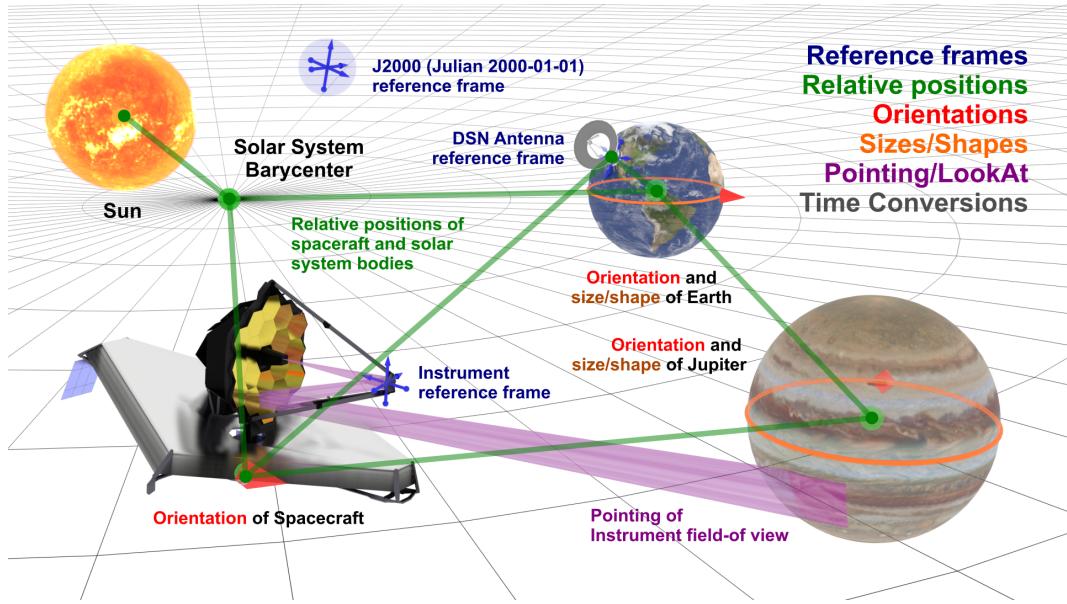


Figure 5. NASA SPICE

A tutorial presentation about SPICE is available [here](#).

The [Frames Kernel](#) is the key component of SPICE to link reference frames and which in particular inspired the frame transformations in OGC GeoPose.

### Space Data Standards

International space data standards are documented in [Consultative Committee for Space Data Systems \(CCSDS\) Blue Books](#). Spacecraft position and orientation are described as *attitude* as described in section 5.3 of [CCSDS Navigation Data - Definitions and Conventions](#). Typical GeoPose use cases include antenna tracking, sun sensor, star sensor, gyro package and horizon sensor.

Beyond those mentioned in this section, there are other [space data standards](#) with which the reviewer may wish to be acquainted.

# Chapter 7. Conclusion

Thank you for taking the time to read this guide for OGC GeoPose reviewers. Please visit the [GeoPose website](#) for further details about implementers and other supporting tools.

Any errors or omissions can be corrected upon review. If you would like to suggest changes to this guide, please create a new issue in the [Reviewers Guide GitHub repository](#).

If you have questions or suggestions for the GeoPose specification, we also request that an issue be created. Issues or feedback about the standard can be created in the [GeoPose GitHub repository](#).

To sign up for periodic announcements and updates about GeoPose, join our [mailing list](#).

If there are any other issues, our [contact form](#) will reach the SWG co-chairs and members.

# Annex A: Revision History

Date	Release	Editor	Primary clauses modified	Description
2021-06-15	0.1	C.Perey and J. Morley	all	initial version
2021-09-30	0.5	C.Perey and J. Morley	all	initial version
2022-01-21	0.6	GeoPose SWG	all	draft version
2022-02-07	0.7	C.Perey	all	draft version
2022-02-10	0.8	C.Perey	7.4	draft final version
2022-02-11	0.9	C.Perey	7.4	draft final version