# Astronomical Coordinates and Coordinate Systems

# Version 1.0

## IVOA Working Draft 2019-03-20

Working group
Data Model Working Group
This version
http://www.ivoa.net/documents/Coordinates/20190320
Latest version
http://www.ivoa.net/documents/Coordinates
Previous versions
This is the first public release
Author(s)
Arnold Rots, Mark Cresitello-Dittmar, Omar Laurino
Editor(s)
Arnold Rots, Mark Cresitello-Dittmar

## Abstract

In creating version 2 of the "Space-Time Coordinate Metadata for the Virtual Observatory" (STC) Data Model (Rots, 2007), it was decided to split the content into various component models which focus on particular aspects of the previous model scope.

This model describes the Coordinates model and covers the following concepts.

- Description of single and multi-dimensional coordinate space, and coordinates within that space.
- Coordinate Frames, providing metadata describing the origin and orientation of the coordinate space.
- Definition of simple domain-specific coordinate types for the most common use cases.
- Coordinate Systems, a collection of coordinate frames.

## Status of this document



## Contents

## Acknowledgments

This document has been developed with support from NSF and NASA under the Virtual Astronomical Observatory (VAO) project, the National Science Foundation's (http://www.nsf.gov) Information Technology Research Program under Cooperative Agreement AST0122449 with The Johns Hopkins University, from the UK Particle Physics and Astronomy Research Council (PPARC),http://www.pparc.ac.uk, and from the Euro-VO projects (European Commission 7th program): Euro-VO Aida, VO-ICE and CoSADIE.

## Conformance-related definitions

The words "MUST", "SHALL", "SHOULD", "MAY", "RECOMMENDED", and "OPTIONAL" (in upper or lower case) used in this document are to be interpreted as described in IETF standard RFC2119 (Bradner, 1997).

The *Virtual Observatory (VO)* is a general term for a collection of federated resources that can be used to conduct astronomical research, education, and outreach. The International Virtual Observatory Alliance (IVOA) is a global collaboration of separately funded projects to develop standards and infrastructure that enable VO applications.

# 1 Introduction

## 1.1 Motivation

Astronomy, being primarily a science that crucially depends on observations, has a very basic need for complete, accurate, and unambiguous metadata regarding coordinate information, meaning all coordinates of the observable space and noting that several of these are intertwined. The Data Model described in this document aims to provide a model for such metadata.

## 1.2 Context and Scope

This document results from updating the "Space-Time Coordinate Metadata for the Virtual Observatory" (STC) (Rots, 2007) model for use in VO-DML compliant models. That model provides metadata describing Space-Time related, and other Coordinates. These metadata are to be used for specifying coordinate-related information for datasets, catalogs, and queries.

The update and revision of the STC model has sub-divided the content into component models, each covering a portion of the scope of the parent model. This allows for a better description of the relations between the various components, allows for independent development of the component models, and creates smaller, more digestible content for users.

This document describes the Coordinates model which provides the metadata describing:

- the coordinate space; axes, and domain ranges
- coordinate frames with metadata describing the origin and orientation of the coordinate space
- a general model for specifying coordinate values within the coordinate space
- simple, specialized coordinates for the most common use cases
- coordinate systems associating related coordinate frames

# 2 Use Cases and Requirements

## 2.1 Use Cases

### 2.1.1 Cube model support

The primary use case for this work is in support of the CubesDM.
The CubesDM is a N-Dimensional model for pixelated images and sparse cube data. The following is a brief outline of the most relevant features pertaining to the development of the Measurement, Coordinates, and Transform component models.

- General
  - knowledge of the pixel and physical domain spaces provided at a high level
  - definition of the domain space includes the following criteria
    - dimensionality (typically 1,2 or 3 for physical domain), pixel domain may be of any dimension
    - axis configuration (for spatial domain which has >1D). The most common configurations for astronomical data are Cartesian and Spherical, but others may be used as well.

    * domain range along each axis, typically +/- Inf, but may be limited due to physical constraints (e.g. physical size of a detector, sensitivity limitiations, etc)
    * association with additional metadata further describing the nature of the domain space ( Frame ). This is especially true for the Spatial and Temporal domains, but may apply to others as well. Examples include:
      · reference position (location of origin)
      · reference frame (orientation of the domain space)
      · planetary ephemeris
      · equinox

- Pixelated Image Cube
  - complete specification of pixel coordinate domain; number of axes, number of pixels per axis
  - mappings of various pixel axes to corresponding physical axes
    * spatial domain typically mapping 2-3 pixel axes to physical axes
    * other domains are typically 1 dimensional
    * pixel axes may be involved in multiple mappings to different physical spaces
  - mappings may be stacked to define progressive transitions through a domain (e.g.: pixel -> ccd -> detector -> sky -> wcs )
    * intermediate stages may or may not be explicitely defined
  - image data value is typically given in a physical domain, but may itself be mapped to other domains

- Sparse Cube
  - data axes cover a wide array of physical domains including, but not limited to Spatial, Temporal, Spectral, Polarization,
  - individual domains may be represented multiple times in different frames ( ccd, detector, sky; pha, energy )
  - data values may have associated errors
    * typical error forms include: symmetric( +/- a ), asymmetric( +a:-b ), interval ( a:b ), matrix, elliptical
    * can become quite complex: probability distributions, error maps, etc.
    * quality indicators:
      · global status, typically numeric
      · bit array, where each bit is associated with a particular quality state
    * associated errors may be separable or correlated among multiple data axes
  - data axes may be virtual, defined as a mapping from other data axes (same description as above)

- Physical Data (Observables)
  - focus on the following domains which are frequently included in astronomical data cubes: Spatial, Spectral, Temporal, Polarization
  - Spatial
    * Cartesian space: chip, detector, sky

        * Spherical space: Equatorial, Ecliptic, Galactic, LongLat
    - Time
        * 1-Dimensional: JD, MJD, ISOTime, TimeOffset
    - Polarization
        * Discrete space: Polarization states (Stokes, Linear, Circular, Vector )
    - Spectral
        * 1-Dimensional: energy, frequency, wavelength

### 2.1.2 Transform workflow

An implementation project focused on the Transform model. The purpose of which is to exercise the Transform model through a workflow consisting of:

- serialization in YAML of various Transform operation sequences
- the generation and passing thereof between two Transform library implementations ( AST, gWCS )

## 2.2 Requirements

Examination and implementation of the above cases leads to a set of requirements distributed through the various STC component models. Here we itemize those relevant to the coordinates model specifically. We note that some elements of this model are included based on the knowledge and experience incorporated into previous STC data model and not from direct requirements from the given use cases.

In Appendix A, we provide a mapping of the data model elements to the various requirements they serve.

### 2.2.1 General

Requirements pertaining to the overall criteria that the model must satisfy.

**[vodml.001]:** The model shall be vo-dml compliant

**[vodml.002]:** shall re-use, or refer to, dependent models for objects and concepts already defined in other models

**[vodml.003]:** shall produce a validated vo-dml XML description

**[vodml.004]:** shall produce documentation in vo-dml HTML format

**[vodml.005]:** shall produce documentation in standard PDF format

### 2.2.2 Application/Usage

Requirements pertaining to the user experience. Note, as a data model, users will not typically interact directly with the model,

**[user.001]:** Users should be able to identify and use basic content with minimal specialized information. In other words, a generic utility should be able to find and use core elements without knowing a lot about the various extensions and uses of those elements.

**[user.002]:** When applicable, the model should support usability by simplifying common scenarios. i.e. common things simple, complex things possible

### 2.2.3 Content

Requirements pertaining to the elements to be defined by the model.

- Domains

  **[dom.001]:** Shall accommodate the description of data in any observable domain

  **[dom.002]:** Shall provide enhanced/specialized description for data pertaining to

  - **[dom.002.1]:** Pixel domain: binned, integerized, n-dimensional domain
  - **[dom.002.2]:** Spatial domain: continuous domain, typically in 2-3 dimensional cartesian or spherical spaces
  - **[dom.002.3]:** Time domain: continuous 1D domain, typically provided in JD, MJD, ISO, or as an Offset from a zero point
  - **[dom.002.4]:** Polarization domain: discrete 1D domain of polarization states.

- Coordinates

  - Coordinate Spaces:

    **[coords.001]:** Shall facilitate the description of the domain space

    - **[coords.001.1]:** Coordinate space shall consist of 1 to N dimensional axes
    - **[coords.001.2]:** Shall support the description of axes which are continuous, binned, and discrete in nature
    - **[coords.001.3]:** Each dimensional axis shall define the domain range of that axis as appropriate for its nature

  - Coordinate frames:

    **[coords.002]:** Shall facilitate the specification of the nature of the domain, providing additional metadata relevant to the interpretation of coordinates in that domain.

  - Coordinates:

    **[coords.003]:** Shall identify a location within the coordinate domain space

    **[coords.004]:** Shall be associated with a corresponding coordinate frame providing metadata relevant to the interpretation of the coordinate

    **[coords.005]:** Shall be associated with a particular axis of the coordinate space to provide context for the coordinate and facilitate the application of mapping Transforms

    **[coords.006]:** Shall be complete quantities, including value and units as appropriate

    **[coords.007]:** Shall support the association of atomic coordinates into a multi-dimensional compound grouping

  - Coordinate systems:

    **[coords.008]:** Shall provide for encapsulating the description of the entire domain space

    **[coords.009]:** for Pixel domain, this must include the full coordinate space description

    **[coords.010]:** for Physical domains, this must include the Frame specifications, as it is this metadata that is more relevant to users. The coordinate space is typically well defined or implied by the coordinate itself.

## 2.3 Role within the VO Architecture

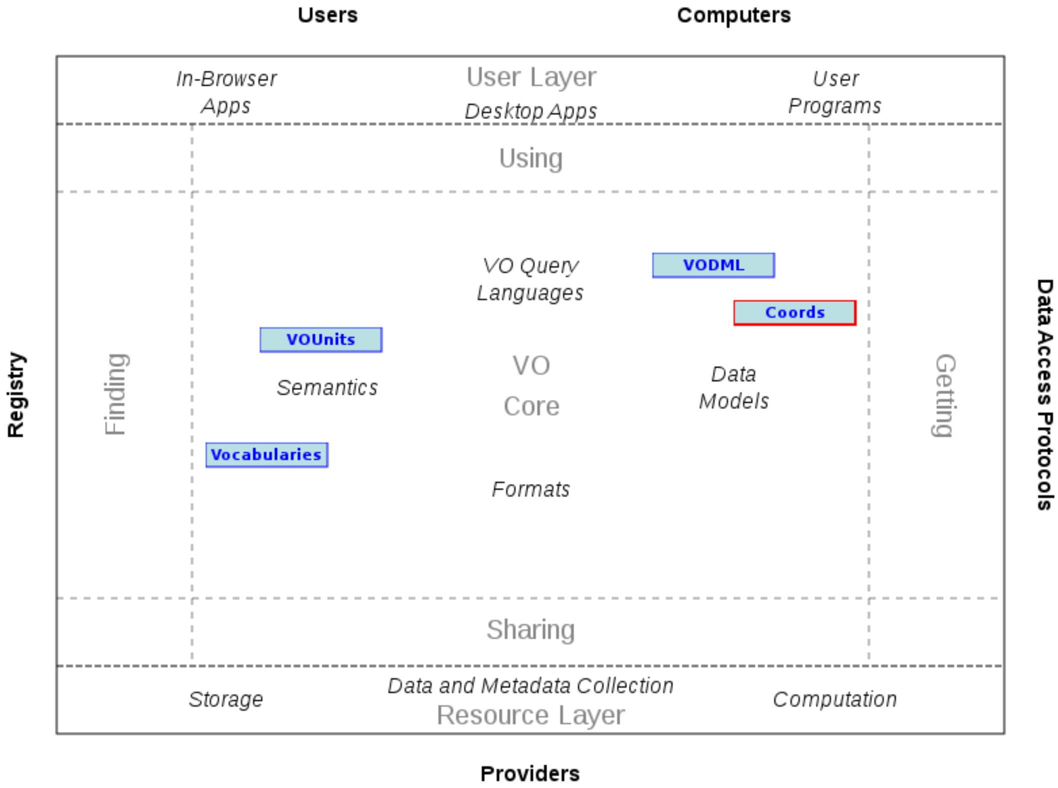



*Figure 1:* Architecture diagram for this document

Fig. 1 shows the role this document plays within the IVOA architecture (Arviset and Gaudet et al., 2010).

# 3 Model: coords

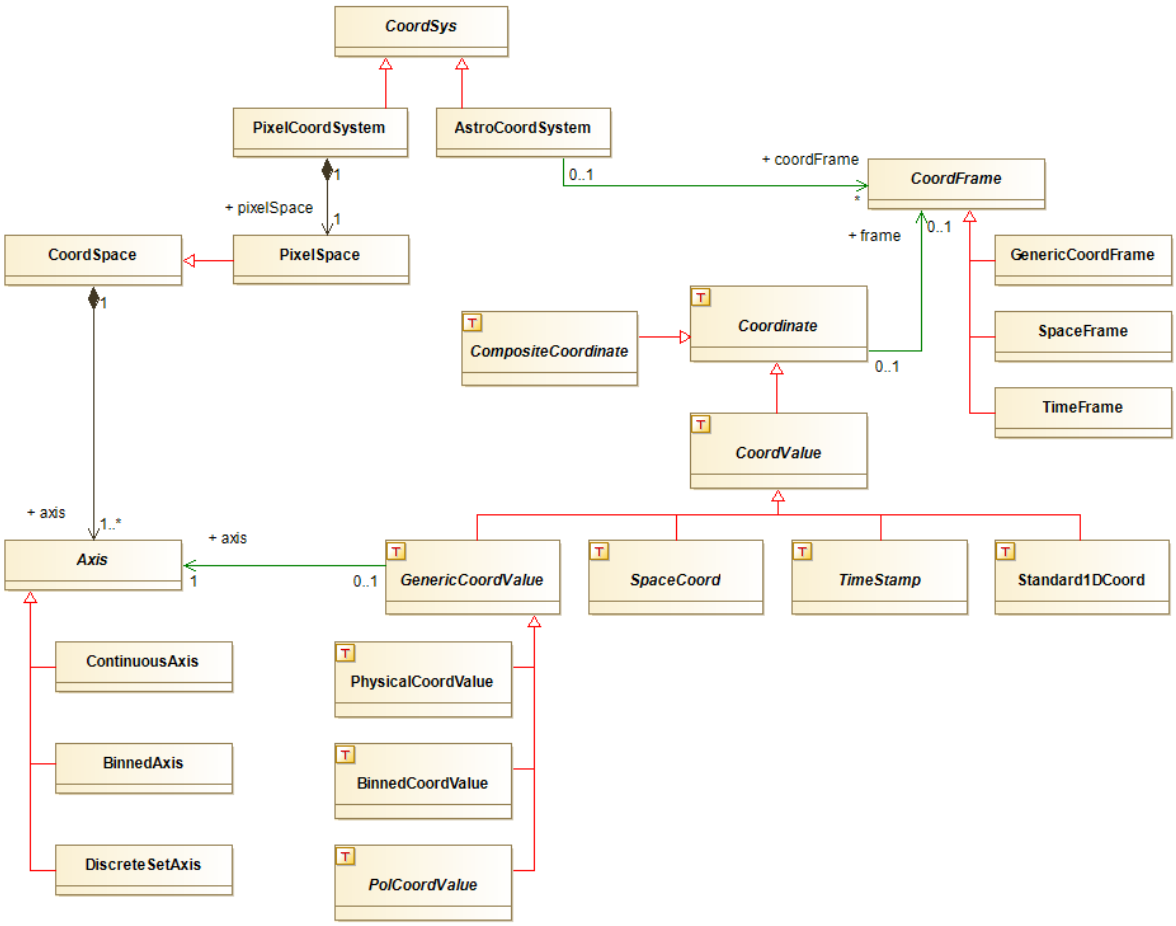



*Figure 2:* Model Overview

This model defines objects which describe the coordinate space, coordinates within that space, and frames, which provide additional metadata regarding the origin, orientation, etc, of the coordinate space. The model also defines a coordinate system, bundling frames into associated groups.

# 4 Coordinates

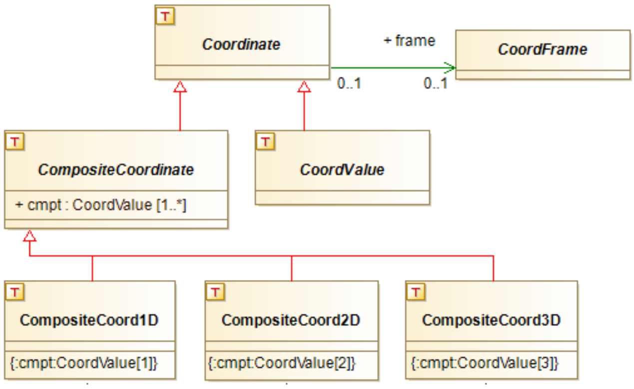



*Figure 3:* Top level Coordinate elements

## 4.1 Coordinate (Abstract)

Abstract base class for the Coordinate data types which represent an absolute location within a coordinate space. Coordinates MAY refer to a coordinate frame, providing additional metadata relevant to interpreting the coordinate value.

### 4.1.1 Coordinate.frame

**vodml-id: Coordinate.frame**
**type: coords:CoordFrame**
**multiplicity: 0..1**
Provided additional metadata relevant to interpreting the coordinate value; for example, the spatial reference position, or time scale.

## 4.2 CoordValue (Abstract)

Abstract head of the atomic coordinate value types. Each coordinate value MUST reference the associated axis in the coordinate space. Combined with the inherited coordinate frame reference, the CoordValue type basically represents the phrase "I am a value along that axis in that frame." NOTE: In this model we provide a generic means for describing any sort of data, as well as a set of simple, specialized Coordinate types for the most common data which encapsulate much of the domain space metadata into the definition of the class itself.

## 4.3 CompositeCoordinate (Abstract)

Multi-dimensional coordinate value. This container can be used to collect coordinate values which should be considered as a single entity. Any concrete class of this type MUST contain a specific number of component values in order to comply with vo-dml modeling rules. We define concrete classes for 1-D, 2-D, and 3-D cases. As a Coordinate, this class MAY include a coordinate frame reference, as do the component coordinate values. In practice, the coordinate frame reference may be on either the composite coordinate or the component coordinate values, but MUST NOT be on both.

### 4.3.1 CompositeCoordinate.cmpt

**vodml-id: CompositeCoordinate.cmpt**
**type: coords:CoordValue**
**multiplicity: 1..***
Component member of the composite coordinate. All components MUST refer to axes of the same coordinate space.

## 4.4 CompositeCoord1D

A 1-dimensional composite coordinate. By itself, this type has limited usefulness, but facilitates cases where the coordinate dimensionality may not be known a priori.
**constraint**
  **detail: CompositeCoord1D.cmpt:CoordValue[1]**

## 4.5 CompositeCoord2D

A 2-dimensional composite coordinate.
**constraint**
  **detail: CompositeCoord2D.cmpt:CoordValue[2]**

## 4.6 CompositeCoord3D

A 3-dimensional composite coordinate.
**constraint**
  **detail: CompositeCoord3D.cmpt:CoordValue[3]**

# 5 Coordinate Frames

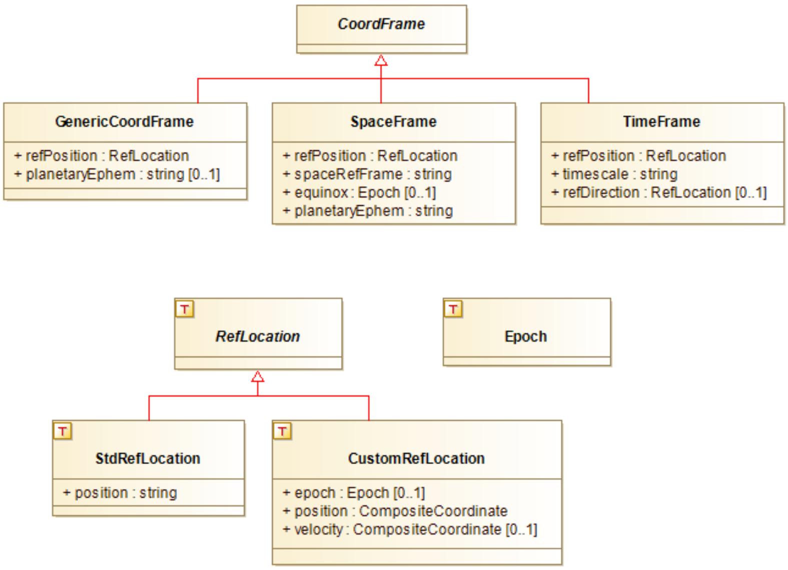



*Figure 4:* Coordinate Frame elements

## 5.1 CoordFrame (Abstract)

This is the abstract, empty, base class for all coordinate frames. Coordinate frames provide metadata associated with the coordinate domain space. Typically, this will be related to the origin and orientation of the axes, but might include any metadata which pertains to the definition of the domain.

## 5.2 GenericCoordFrame

The generic coordinate frame is for cases where a domain-specific frame (e.g.: Space, Time), is not required, but the relevant reference metadata is still needed (e.g.: for Redshift or Spectral data)

### 5.2.1 GenericCoordFrame.refPosition

**vodml-id: GenericCoordFrame.refPosition**
**type: coords:RefLocation**
**multiplicity: 1**
Spatial location in phase space (position and velocity) at which the observed value is considered to have been taken. This will typically be given by a standard reference position, but we allow for custom locations as well.

### 5.2.2 GenericCoordFrame.planetaryEphem

**vodml-id: GenericCoordFrame.planetaryEphem**
**type: ivoa:string**
**multiplicity: 0..1**

A planetary ephemeris MAY be provided, and SHOULD be provided whenever appropriate, to indicate which solar system ephemeris was used. If needed, but not provided, it is assumed to be "DE405"

## 5.3 SpaceFrame

A SpaceFrame is specified by its reference frame (orientation), and a reference position (origin). Currently only standard reference frames are allowed. An equinox MUST be provided for pre-ICRS reference frames. A planetary ephemeris MAY be provided if relevant. If needed, but not provided, it is assumed to be "DE 405".

### 5.3.1 SpaceFrame.refPosition

**vodml-id: SpaceFrame.refPosition**
**type: coords:RefLocation**
**multiplicity: 1**
The spatial location at which the coordinates are considered to have been determined. This model supports locations provided as either a standard reference position (e.g. GEOCENTER), or a coordinate specifying a custom location (e.g. long, lat, height ).

### 5.3.2 SpaceFrame.spaceRefFrame

**vodml-id: SpaceFrame.spaceRefFrame**
**type: ivoa:string**
**vocabulary: https://ivoa.net/vocabularies/coords/SpaceRefFrame**
**multiplicity: 1**
The spatial reference frame. Values MUST be selected from the controlled vocabulary at the given URL.

### 5.3.3 SpaceFrame.equinox

**vodml-id: SpaceFrame.equinox**
**type: coords:Epoch**
**multiplicity: 0..1**
Reference date for the frame, required for pre-ICRS reference frames.

### 5.3.4 SpaceFrame.planetaryEphem

**vodml-id: SpaceFrame.planetaryEphem**
**type: ivoa:string**
**multiplicity: 1**
Ephemeris file for solar system objects SHOULD be specified whenever relevant.

## 5.4 TimeFrame

A TimeFrame SHALL include a time scale and reference position. It MAY also include a reference direction.

#### 5.4.1 TimeFrame.refPosition

**vodml-id: TimeFrame.refPosition**
**type: coords:RefLocation**
**multiplicity: 1**
The spatial location at which the coordinate is considered to have been taken. This model supports locations provided as either a standard reference position (e.g. GEOCENTER), or a coordinate specifying a custom location (e.g. long, lat, height).

#### 5.4.2 TimeFrame.timescale

**vodml-id: TimeFrame.timescale**
**type: ivoa:string**
**vocabulary: https://ivoa.net/vocabularies/coords/TimeScale**
**multiplicity: 1**
The time scale sets the reference frame. The value MUST be selected from the controlled vocabulary at the given URL.

#### 5.4.3 TimeFrame.refDirection

**vodml-id: TimeFrame.refDirection**
**type: coords:RefLocation**
**multiplicity: 0..1**
The reference direction is needed if the time stamps are transformed to a time frame with a different reference position. In those situations, the solar system ephemeris also comes into play. See: FITS WCS Paper IV (Rots and Bunclark et al., 2015) for details, but in short: The reference direction, presumably the direction to the thing being observed, is used in conjunction with the reference position and planetary ephemeris to determine the correction applied for the path length change. To be fully useful, one also needs to know the location at which the observation was made ( i.e. the observatory location), which is not considered to be Frame metadata.

### 5.5 Epoch

We define epoch as a primitive data type with the expected form "<type><year>" where type = "J" or "B" for Julian or Besselian respectively, and year is expressed as a decimal year. e.g.: "B1950", "J2000.0"

### 5.6 RefLocation (Abstract)

RefLocation defines the origin of the spatial coordinate space. This location is represented either by a standard reference position (for which the absolute location in phase space is known by definition), or a specified point in another Spatial frame. This object is used as the origin of the SpaceFrame here, but also to specify the Spatial Reference Position (refPosition) associated with other domain Frames. For example, in the Time domain, the Spatial Reference Position indicates that the 'time' values are the time that the 'event' occured at that location, which might be different from the detector location.

## 5.7 StdRefLocation

An absolute a-priori known location in phase space (position and velocity). Values are selected from the StdRefPosition vocabulary. Considering that the GEOCENTER is really the only place for which we know the absolute location at all times, all other locations require the specification of a planetary ephemeris. LSR[KD] are reserved for spectral and reshift frames. TOPOCENTER (location of the observer) is special in that it assumes that the observing location is available through other means (e.g. a geographic location or an orbit ephemeris). RELOCATABLE is available for simulations. UNKNOWN should only be used if absolutely necessary.

### 5.7.1 StdRefLocation.position

**vodml-id: StdRefLocation.position**
**type: ivoa:string**
**vocabulary: https://ivoa.net/vocabularies/coords/ReferencePosition**
**multiplicity: 1**
Standard reference location. Values MUST be selected from the controlled vocabulary at the given URL.

## 5.8 CustomRefLocation

A custom reference location in phase space (position and velocity). Position and velocity are given as coordinates with an associated SpaceFrame. An epoch MAY be provided to further refine the location.

### 5.8.1 CustomRefLocation.epoch

**vodml-id: CustomRefLocation.epoch**
**type: coords:Epoch**
**multiplicity: 0..1**
Epoch for the reference location.

### 5.8.2 CustomRefLocation.position

**vodml-id: CustomRefLocation.position**
**type: coords:CompositeCoordinate**
**multiplicity: 1**
The spatial coordinates of the reference location.

### 5.8.3 CustomRefLocation.velocity

**vodml-id: CustomRefLocation.velocity**
**type: coords:CompositeCoordinate**
**multiplicity: 0..1**
The velocity of the reference location.

# 6 Standardized Coordinates

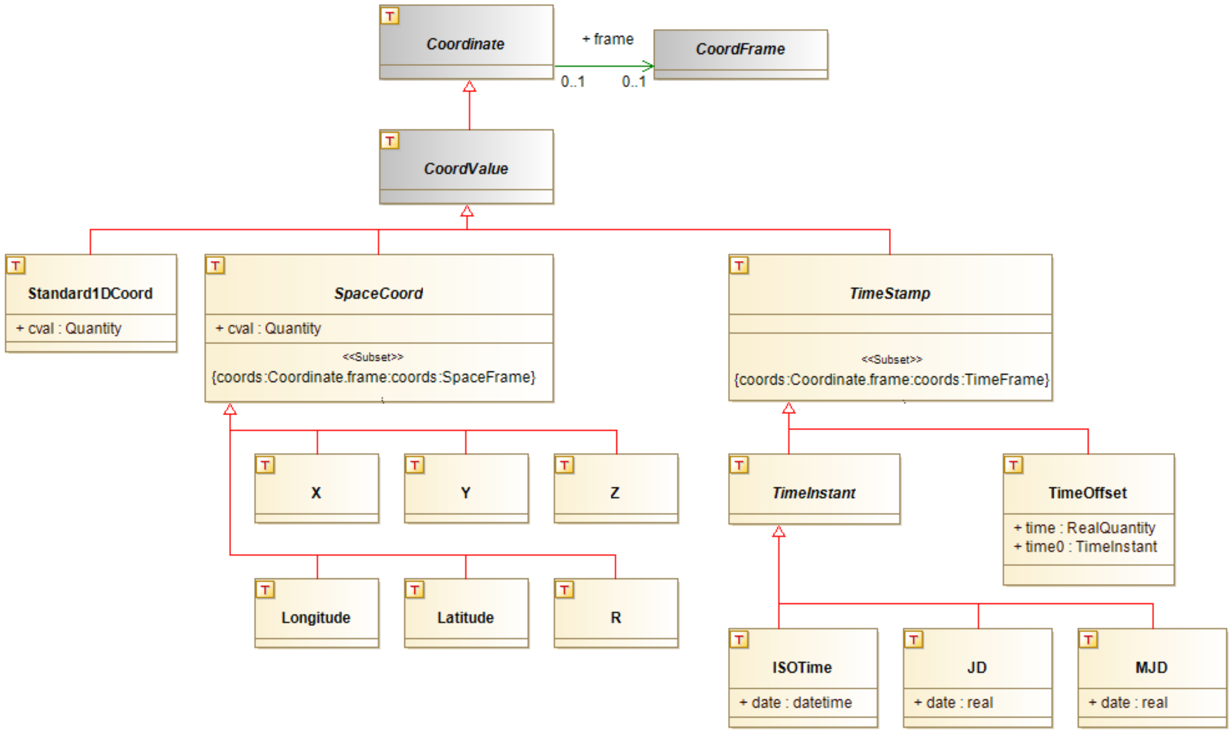



*Figure 5:* Standardized Coordinates

This section provides a set of standardized coordinates for the most commonly used coordinate types. These coordinates incorporate the description of the associated coordinate space in which they reside into their definitions, thereby reducing the need to include 'boilerplate' content. **It is expected that these coordinates will be used in the vast majority of cases.**

A Brief Primer on Time Metadata; for reference and more information, see: FITS WCS Paper IV (Rots and Bunclark et al., 2015).

1. Required:
   * Record time stamps in JD, MJD, ISO-8601, or elapsed time. If in elapsed time, a zero point MUST be given in a time stamp which is not itself an elapsed time.
   * Provide the time scale used (e.g. TT, TDB, TAI, GPS, ET, UTC, TCG, TCB).
   * Provide the reference position (place where the time is measured).

2. Note the following:
   * JD and MJD do not imply a time scale; it needs to be provided separately.
   * JD and MJD are dimensionless, though a unit of 'day' is implied.
   * It is a bad idea to mix UTC with JD or MJD, since not all UTC days are the same length. Instead, use the restricted form of ISO-8601: [[+|-]c]ccyy-mm-dd[Thh[:mm[:ss[.ss...]]]], with no time zone characters.
   * TDB runs on average synchronously with TT, but corrects for the relativistic effects caused by deviations in the orbit of the Earth from perfect circularity and constant gravitational potential.

3. Recommendations:
   * Avoid UTC. It is trivial to convert the times provided by, e.g., space agencies, to TT immediately when you get them, and it will save headaches later on.
   * Use TT: it is the official IAU time scale, continuous with ET and the one which solar system ephemerides are based upon.
   * TAI and GPS are acceptable alternatives, with constant offsets from TT.
   * Use the same reference position for time and space and make sure it is commensurate with your time scale. For instance, when you convert to the barycenter, also convert to TDB.
   * Be aware that the barycenter is not the heliocenter
   * Be specific in labeling the time axis; e.g.: JD(TT;GEOCENTER) or MJD(TDB; BARYCENTER).
   * Use proleptic Gregorian dates for ISO-8601.

4. Never use:
   * TJD, HJD, BJD, etc. These are not officially recognized and suggest certain metadata values, but leave considerable ambiguity as to what those metadata values actually are. Instead, specify your metadata explicitly. It avoids confusion later on and is not much more work.

5. What if you deal with incomplete data?
   * If you do not know the time scale and/or reference position, you can provide them as UNKNOWN and set the systematic error/uncertainty to, say, 1000 s. 100 s will do if only the time scale is unknown.

6. What else is there to know?
   * Quite a lot, especially the so-called coordinate time scales (TCG and TCB). Because TDB runs, on average, synchronously with TT, but in a very different potential well, which requires different values for fundamental physical constants in the barycenter. That is awkward and the coordinate time scales fix that by running at different rates. More in the above cited A&A paper.

## 6.1 Standard1DCoord

Standardized coordinate appropriate for any one-dimensional coordinate within a standard 1D coordinate space, that is not represented by another type. For example, Temperature, Pressure, Magnitude, Energy, etc.

### 6.1.1 Standard1DCoord.cval

**vodml-id: Standard1DCoord.cval**
**type: ivoa:Quantity**
**multiplicity: 1**
Coordinate value along the axis.

## 6.2 SpaceCoord (Abstract)

Abstract head of a set of specialized coordinates for the spatial domain which cover the most commonly used cases. All SpaceCoords SHOULD refer to an appropriate SpaceFrame. In this

model, we define 2 standard spatial coordinate space instances, (Cartesian and Spherical, see Appendix B ). Here, we provide atomic coordinates associated with these spaces which can be used in a variety of different applications.
**subset**
  **role: coords:Coordinate.frame**
  **type: coords:SpaceFrame**

#### 6.2.1 SpaceCoord.cval

**vodml-id: SpaceCoord.cval**
**type: ivoa:Quantity**
**multiplicity: 1**
Coordinate value along the designated axis.

### 6.3 X

A spatial location along the 'X' axis of a standard 3D Cartesian coordinate space (B.1).

### 6.4 Y

A spatial location along the 'Y' axis of a standard 3D Cartesian coordinate space (B.1).

### 6.5 Z

A spatial location along the 'Z' axis of a standard 3D Cartesian coordinate space (B.1).

### 6.6 Longitude

A spatial location along the 'Longitude' axis of a standard 3D Spherical coordinate space (B.2).

### 6.7 Latitude

A spatial location along the 'Latitude' axis of a standard 3D Spherical coordinate space (B.2).

### 6.8 R

A spatial location along the radial axis of a standard 3D Spherical coordinate space (B.2).

### 6.9 TimeStamp (Abstract)

This is the abstract basis for a set of simple time domain coordinates which are expected to accommodate the vast majority of use cases. All TimeStamps, by definition, refer to the axis of a standard 1-D coordinate space, with domainMin|Max of +/-Infinity. As such, there is no 'axis' reference on TimeStamps. All TimeStamps SHOULD refer to an appropriate TimeFrame.
**subset**
  **role: coords:Coordinate.frame**
  **type: coords:TimeFrame**

## 6.10 TimeInstant (Abstract)

TimeStamps which specify a specific instant in time. We define three subtypes (ISOTime, JD, MJD), which allow users to explicitly identify the representation and interpretation of the TimeInstant.

## 6.11 ISOTime

Extension of TimeInstant for time expressed as a structured datetime string. The string representation of a datetime value should follow the FITS convention for representing dates (Hanisch and Farris et al., 2001). The FITS standard is effectively ISO8601 format without the 'Z' tag to indicate UTC: YYYY-MM-DD['T'hh:mm:ss[.SSS]]. The TimeScale is provided in the associated TimeFrame.

### 6.11.1 ISOTime.date

**vodml-id: ISOTime.date**
**type: ivoa:datetime**
**multiplicity: 1**
The ISOTime coordinate value.

## 6.12 JD

Extension of TimeInstant for time expressed in Julian days. Note that JD does not properly specify a time stamp unless it is related to a time scale and reference position. Precision can easily become an issue with JD, as the numbers tend to be large.

### 6.12.1 JD.date

**vodml-id: JD.date**
**type: ivoa:real**
**multiplicity: 1**
The JD coordinate value. JD dates are dimensionless, with implied units in days.

## 6.13 MJD

Extension of TimeInstant for time expressed in Modified Julian Days. T(MJD) = T(JD) - 2400000.5.

### 6.13.1 MJD.date

**vodml-id: MJD.date**
**type: ivoa:real**
**multiplicity: 1**
The MJD coordinate value. MJD dates are dimensionless, with implied units in days.

## 6.14 TimeOffset

Time is given as an offset from a specific point in time (time0).

#### 6.14.1 TimeOffset.time

**vodml-id: TimeOffset.time**
**type: ivoa:RealQuantity**
**multiplicity: 1**
The TimeOffset coordinate value.

#### 6.14.2 TimeOffset.time0

**vodml-id: TimeOffset.time0**
**type: coords:TimeInstant**
**multiplicity: 1**
The reference time from which the offset is calculated. This MUST be given as a TimeInstant (e.g.: JD, MJD, ISOTime).

# 7 Generic Coordinates

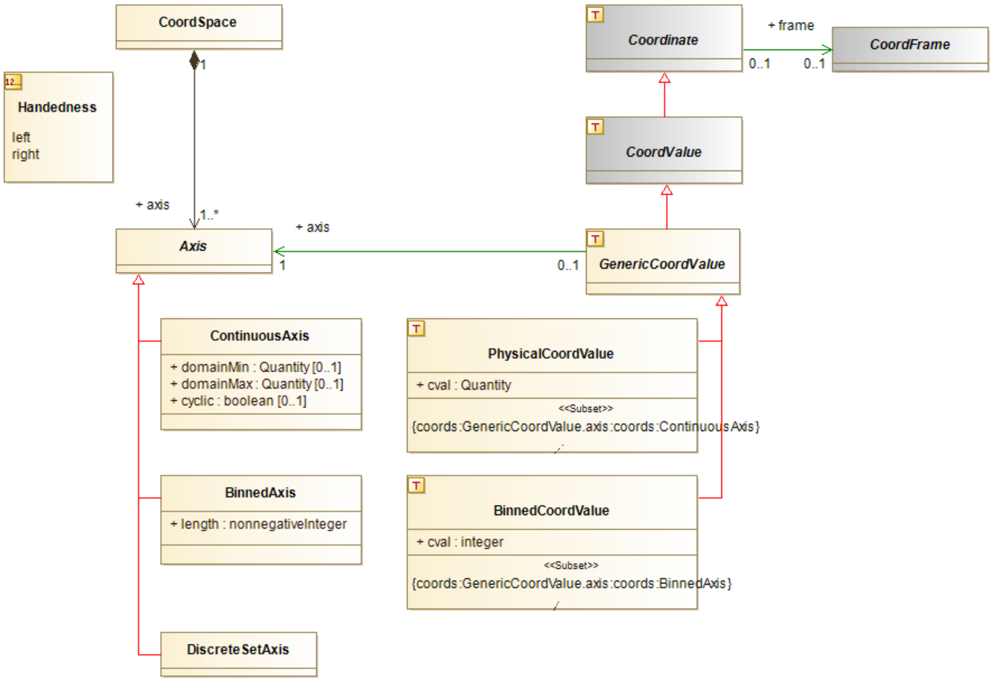



*Figure 6:* Generic Coordinates

For cases where standardized coordinates are not applicable:

- finite domain spaces: detector coordinates, pulse height amplitude, pixellated images, etc.
- non-standardized spaces: MSC(theta,phi) - off axis angle, azimuth
- discrete data: Polarization

a more complete description of the coordinate space is needed. Here, we provide a generic model to describe virtually any form of data.

## 7.1 CoordSpace

This object defines a domain space. i.e.: it describes the set of possible coordinate values.

### 7.1.1 CoordSpace.axis

**vodml-id: CoordSpace.axis**
**type: coords:Axis**
**multiplicity: 1..***
Describes an axis of the coordinate space.

## 7.2 Axis (Abstract)

The abstract parent class for all coordinate axis types. We provide concrete classes for the most common types of data, Continuous, Binned, and Discrete, but allow extension for other types as needed.

### 7.2.1 Axis.name

**vodml-id: Axis.name**
**type: ivoa:string**
**multiplicity: 0..1**
Freeform string, provides the name or label for the axis.

## 7.3 ContinuousAxis

Axis description for continuous data. This object describes the domain for a particular axis of the domain space. It allows for the specification of the legal domain range (min,max), and a flag indicating if the axis is cyclic.

### 7.3.1 ContinuousAxis.domainMin

**vodml-id: ContinuousAxis.domainMin**
**type: ivoa:Quantity**
**multiplicity: 0..1**
Minimum extent of the axis domain space. If not provided, the domain space is considered to have no lower bound (-INFINITY).

### 7.3.2 ContinuousAxis.domainMax

**vodml-id: ContinuousAxis.domainMax**
**type: ivoa:Quantity**
**multiplicity: 0..1**
Maximum extent of the axis domain space. If not provided, the domain space is considered to have no upper bound (+INFINITY).

### 7.3.3 ContinuousAxis.cyclic

**vodml-id: ContinuousAxis.cyclic**
**type: ivoa:boolean**
**multiplicity: 0..1**
Flag indicating if the axis is cyclic in nature. If not provided, it is assumed to be FALSE.

## 7.4 BinnedAxis

Axis description for binned data, where values along the axis correspond to a bin number.

### 7.4.1 BinnedAxis.length

**vodml-id: BinnedAxis.length**
**type: ivoa:nonnegativeInteger**
**multiplicity: 1**
The length, or number of bins, along the axis.

### 7.5 DiscreteSetAxis

Axis type specifically intended for enumerated coordinates. Since the content and nature of this axis type is heavily dependent on the use case, we define no additional metadata here. Extensions of this type may include additional metadata relevant to the particular use cases. For example, an extension could include the allowed set of values.

### 7.6 Handedness

The handedness of a coordinate space. For most cases, this will be a fixed value in the specification of the coordinate space. We provide this element to allow this flexibility when needed. In this document, it is used in the Pixel domain.
Enumeration Literals

**left** : **vodml-id:** Handedness.left
**description:** positive x and y axes point right and up, the positive z axis points inward

**right** : **vodml-id:** Handedness.right
**description:** positive x and y axes point right and up, the positive z axis points outward

### 7.7 GenericCoordValue (Abstract)

Parent for all generic coordinate types. Associates the coordinate with a particular axis of the coordinate space.

#### 7.7.1 GenericCoordValue.axis

**vodml-id: GenericCoordValue.axis**
**type: coords:Axis**
**multiplicity: 1**
Reference to the particular axis of the coordinate space along which this value is given. e.g.: the X axis of a 3D CARTESIAN coordinate space.

### 7.8 PhysicalCoordValue

The most common type of coordinate value. This type is appropriate for any data whose values can be described by an ivoa:Quantity (numeric, with unit).
**subset**
**role: coords:GenericCoordValue.axis**
**type: coords:ContinuousAxis**

#### 7.8.1 PhysicalCoordValue.cval

**vodml-id: PhysicalCoordValue.cval**
**type: ivoa:Quantity**
**multiplicity: 1**
This coordinate MUST contain a value expressed as an ivoa:Quantity.

## 7.9 BinnedCoordValue

Coordinate value type specifically intended for binned data (e.g.: pixel indexes).
**subset**
  **role: coords:GenericCoordValue.axis**
  **type: coords:BinnedAxis**

### 7.9.1 BinnedCoordValue.cval

**vodml-id: BinnedCoordValue.cval**
**type: ivoa:integer**
**multiplicity: 1**
The binned coordinate value, expressed as an integer. e.g.: bin number, pixel index.

# 8 Pixel Coordinates

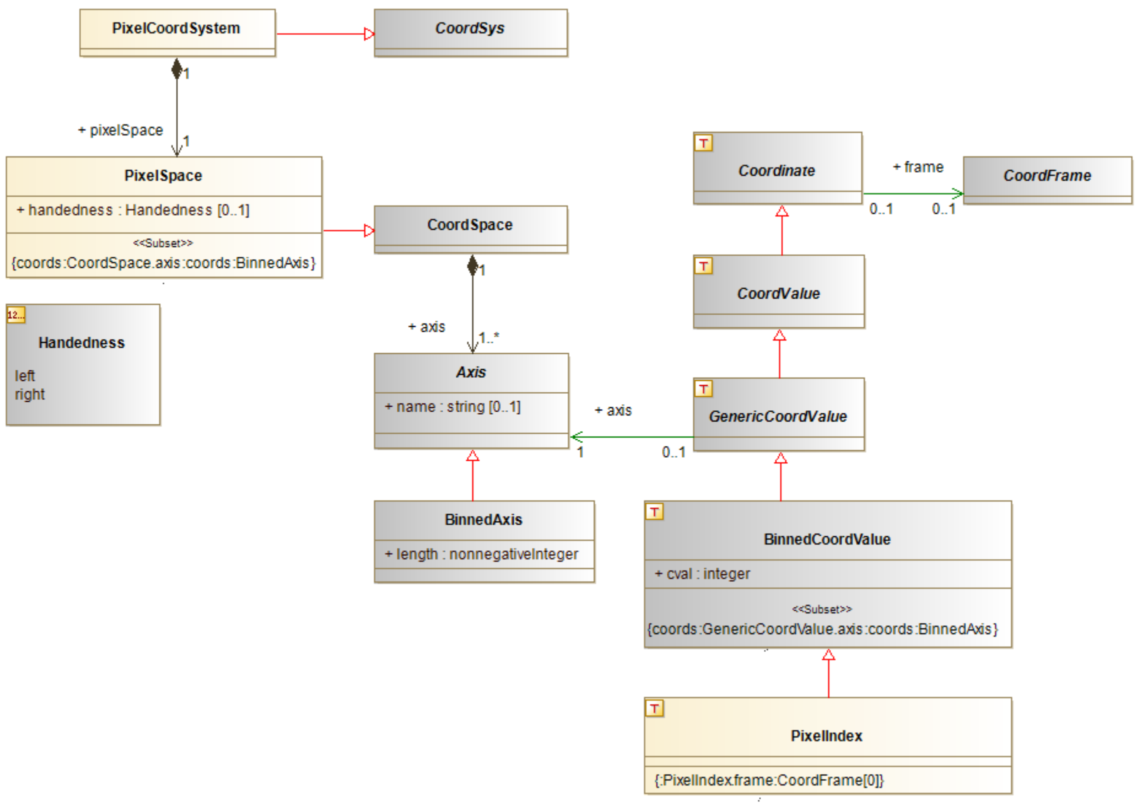



*Figure 7:* Pixel Coordinates

This section extends the core elements to provide specialized content for the Pixel domain. The Pixel coordinate space is defined as a 'virtual' binned space, with no physical meaning. The axes in this space provide integer indices into that space.

## 8.1 PixelCoordSystem

The PixelCoordSystem provides a complete description of the pixel coordinate space. It SHALL contain one PixelSpace instance describing each pixel axis.

### 8.1.1 PixelCoordSystem.pixelSpace

**vodml-id: PixelCoordSystem.pixelSpace**
**type: coords:PixelSpace**
**multiplicity: 1**
The pixel space completely defines the pixel coordinate axes. Each axis MUST be defined as a BinnedAxis type.

## 8.2 PixelSpace

A PixelSpace SHALL include one or more BinnedAxis objects describing the pixel coordinate space. A handedness value MAY be provided to specify the relative orientation of the axes.

**subset**
  **role: coords:CoordSpace.axis**
  **type: coords:BinnedAxis**

### 8.2.1 PixelSpace.handedness

**vodml-id: PixelSpace.handedness**
**type: coords:Handedness**
**multiplicity: 0..1**
Specifies the handedness of the coordinate space.

## 8.3 PixelIndex

A coordinate value in the pixel domain. A 1-dimensional pixel index. There is no frame in the pixel domain, so no frame reference is allowed.
**constraint**
  **detail: PixelIndex.PixelIndex.frame:CoordFrame[0]**

# 9 Polarization Coordinates

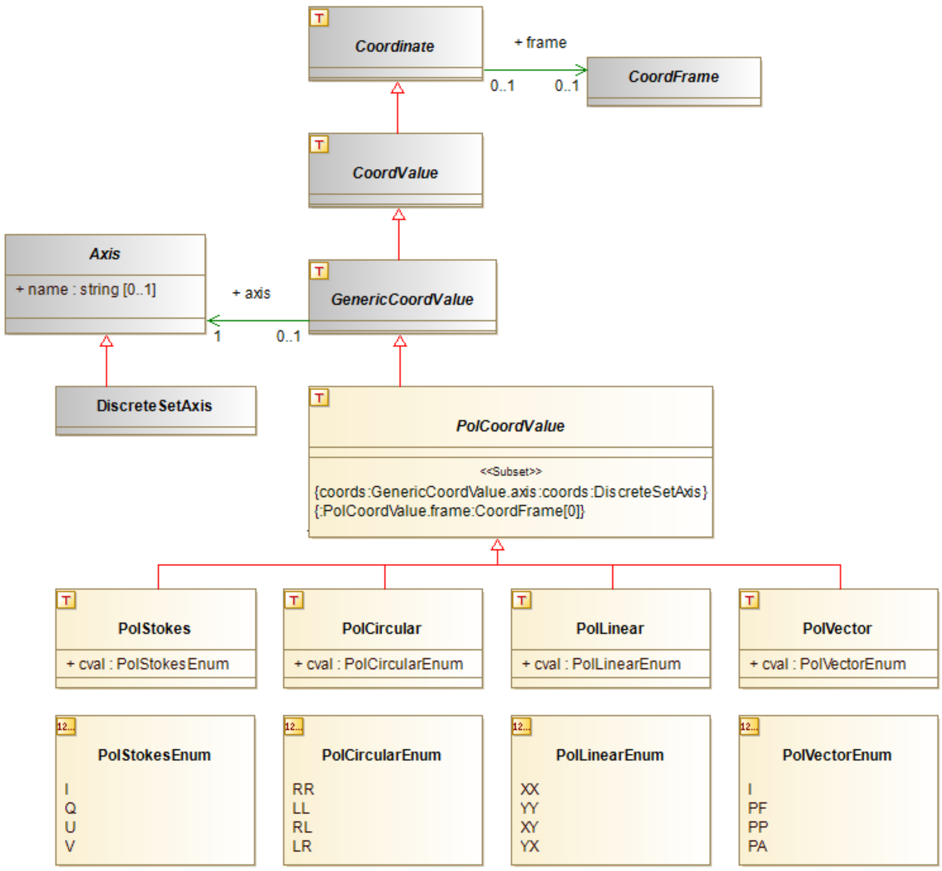



*Figure 8:* Pixel Coordinates

This section extends the core elements to provide specialized content for the Polarization data. Polarization data is modeled as a set of enumerated polarization states. As such, values are associated with a discrete set axis type, providing a concrete example of a Discrete coordinate space usage.

## 9.1 PolCoordValue (Abstract)

Abstract head of the polarization coordinate types. Here we constrain the coordinate value to refer to a discrete axis type.

**subset**

  **role: coords:GenericCoordValue.axis**
  **type: coords:DiscreteSetAxis**

**constraint**

  **detail: PolCoordValue.PolCoordValue.frame:CoordFrame[0]**

## 9.2 PolStokes

Coordinate for Stokes Polarization type

### 9.2.1 PolStokes.cval

**vodml-id: PolStokes.cval**
**type: coords:PolStokesEnum**
**multiplicity: 1**
The coordinate value MUST be from the PolStokesEnum enumerated set.

## 9.3 PolCircular

Coordinate for Circular Polarization type

### 9.3.1 PolCircular.cval

**vodml-id: PolCircular.cval**
**type: coords:PolCircularEnum**
**multiplicity: 1**
The coordinate value MUST be from the PolCircularEnum enumerated set.

## 9.4 PolLinear

Coordinate for LinearPolarization type

### 9.4.1 PolLinear.cval

**vodml-id: PolLinear.cval**
**type: coords:PolLinearEnum**
**multiplicity: 1**
The coordinate value MUST be from the PolLinearEnum enumerated set.

## 9.5 PolVector

Coordinate for Vector Polarization type

### 9.5.1 PolVector.cval

**vodml-id: PolVector.cval**
**type: coords:PolVectorEnum**
**multiplicity: 1**
The coordinate value MUST be from the PolVectorEnum enumerated set.

## 9.6 PolStokesEnum

Stokes Polarization states
Enumeration Literals

- **I** : **vodml-id:** PolStokesEnum.I
- **Q** : **vodml-id:** PolStokesEnum.Q
- **U** : **vodml-id:** PolStokesEnum.U

**V** : **vodml-id:** PolStokesEnum.V

## 9.7 PolCircularEnum

Circular Polarization states
Enumeration Literals

**RR** : **vodml-id:** PolCircularEnum.RR

**LL** : **vodml-id:** PolCircularEnum.LL

**RL** : **vodml-id:** PolCircularEnum.RL

**LR** : **vodml-id:** PolCircularEnum.LR

## 9.8 PolLinearEnum

Linear Polarization states
Enumeration Literals

**XX** : **vodml-id:** PolLinearEnum.XX

**YY** : **vodml-id:** PolLinearEnum.YY

**XY** : **vodml-id:** PolLinearEnum.XY

**YX** : **vodml-id:** PolLinearEnum.YX

## 9.9 PolVectorEnum

Vector Polarization states
Enumeration Literals

**I** : **vodml-id:** PolVectorEnum.I

**PF** : **vodml-id:** PolVectorEnum.PF

**PP** : **vodml-id:** PolVectorEnum.PP

**PA** : **vodml-id:** PolVectorEnum.PA

# 10 Coordinate Systems

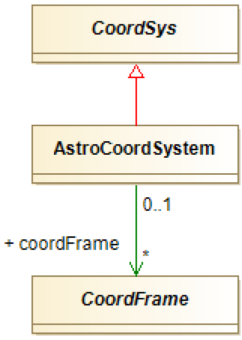



*Figure 9:* Coordinate Systems

## 10.1 CoordSys (Abstract)

The CoordSys object is the parent of a set of containers for organizing Coordinate Frames into related groupings.

## 10.2 AstroCoordSystem

AstroCoordSystem is a container object for organizing physical Coordinate Frame specifications into related groupings. An AstroCoordSystem MAY reference any number of coordinate frames.

### 10.2.1 AstroCoordSystem.coordFrame

**vodml-id: AstroCoordSystem.coordFrame**
**type: coords:CoordFrame**
**multiplicity: 0..***
Frame specification for some domain of the coordinate space.

# A Requirements Mapping

The table below provides a mapping of the various elements of this model to requirements served by that object.

| vodml-id | pertains to |
|---|---|
| AstroCoordSystem | coords.008, coords.010 |
| Axis | coords.001.1, coords.001.2 |
| BinnedAxis | dom.002.1, coords.001.2 |
| BinnedAxis.length | coords.001.3 |
| BinnedCoordValue | dom.001, coords.006 |
| CompositeCoord1D | vodml.001, coords.007 |
| CompositeCoord2D | vodml.001, coords.007 |
| CompositeCoord3D | vodml.001, coords.007 |
| CompositeCoordinate | vodml.001, coords.007 |
| ContinuousAxis | dom.002.2, dom.002.3, coords.001.2 |
| ContinuousAxis.cyclic | coords.001.3 |
| ContinuousAxis.domainMax | coords.001.3 |
| ContinuousAxis.domainMin | coords.001.3 |
| CoordFrame | coords.002, coords.004 |
| CoordSpace | coords.001 |
| CoordSys | coords.008 |
| CoordValue | coords.003, user.001, coords.007 |
| Coordinate | coords.003, coords.007 |
| Coordinate.frame | coords.004, user.001 |
| CustomRefLocation | dom.002.2, coords.002 |
| DiscreteSetAxis | dom.002.4, coords.001.2 |
| Epoch | dom.002.2, coords.001.2 |
| GenericCoordFrame | coords.002 |
| GenericCoordValue | dom.001 |
| GenericCoordValue.axis | coords.005 |
| Handedness | coords.001 |
| ISOTime | user.002, coords.006 |
| JD | user.002, coords.006 |
| Latitude | user.002, dom.002.2, coords.005 |
| Longitude | user.002, dom.002.2, coords.005 |
| MJD | user.002, coords.006 |
| PhysicalCoordValue | dom.001, coords.006 |
| PixelCoordSystem | dom.002.1, coords.008 |
| PixelCoordSystem.pixelSpace | dom.002.1, coords.008, coords.009 |
| PixelIndex | dom.002.1, coords.003, coords.005 |
| PixelSpace | dom.002.1, coords.001.1 |
| PolCircular | dom.002.4 |
| PolCoordValue | dom.001, dom.002.4, coords.003, coords.005 |
| PolLinear | dom.002.4 |
| PolStokes | dom.002.4 |
| PolVector | dom.002.4 |
| R | user.002, dom.002.2, coords.005 |
| RefLocation | dom.002.2, coords.002 |
| SpaceCoord | dom.001, dom.002.2, coords.006, user.001 |
| SpaceFrame | dom.002.2, coords.002 |
| Standard1DCoord | user.002, dom.001, coords.005 |
| StdRefLocation | dom.002.2, coords.002 |
| TimeFrame | dom.002.3, coords.002 |
| TimeInstant | dom.002.3 |
| TimeOffset | user.002, dom.002.3, coords.006 |

| vodml-id | pertains to |
|---|---|
| TimeStamp | user.002, dom.002.3, coords.005, coords.007 |
| X | user.002, dom.002.2, coords.005 |
| Y | user.002, dom.002.2, coords.005 |
| Z | user.002, dom.002.2, coords.005 |

# B Standard Coordinate Spaces

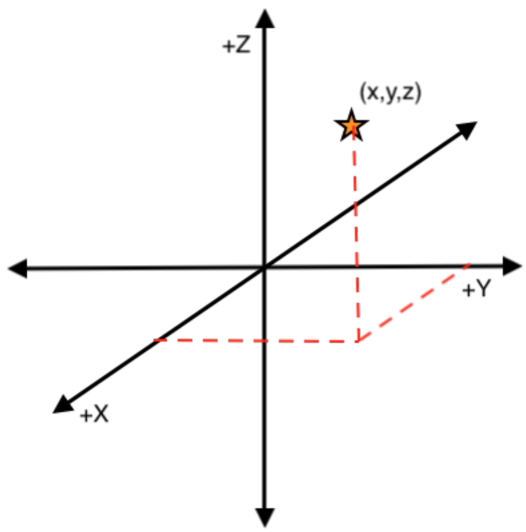
We provide standard instances of commonly used coordinate spaces as instances of elements from this model. Each element has a formal identifier (ID) which can be used to reference the instance given here.

## B.1 Standard Cartesian Coordinate Space

**id**: _CARTESIAN_CoordSpace
Coordinate space comprised of 3 orthogonal axes.
**Axis1**
id: _CARTESIAN_X_Axis
domainMin: -Infinity
domainMax: +Infinity
cyclic: False

**Axis2**
id: _CARTESIAN_Y_Axis
domainMin: -Infinity
domainMax: +Infinity
cyclic: False

**Axis3**
id: _CARTESIAN_Z_Axis
domainMin: -Infinity
domainMax: +Infinity
cyclic: False

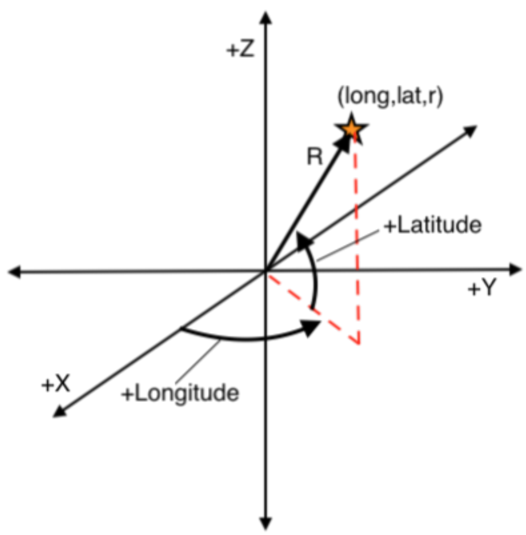



## B.2 Standard Spherical Coordinate Space

**id**: _SPHERICAL_CoordSpace
A 3 dimensional spherical coordinate space, comprised of 2 angular axes and 1 radial axis.
**Axis1**
id: _SPHERICAL_Lat_Axis
domainMin: -90.0 deg
domainMax: +90.0 deg
cyclic: False

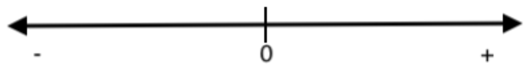
**Axis2**
id: _Spherical_Long_Axis
domainMin: 0.0 deg
domainMax: 360.0 deg
cyclic: True

**Axis3**
id: _Spherical_R_Axis
domainMin: 0.0
domainMax: +Infinity
cyclic: False



## B.3 Standard 1D Coordinate Space

**id**: _STANDARD_1D_CoordSpace
Coordinate space comprised of 1 axis.
**Axis1**
id: Standard_1D_Axis
domainMin: -Infinity
domainMax: +Infinity
cyclic: False



# C Standard Vocabularies

## C.1 Standard Reference Frame (StdRefFrame)

**BaseURL**: http://ivoa.net/vocabularies/coords
**Vocabulary**: ReferenceFrames

We include here the initial Standard Reference Frame Vocabulary. The formal list is stored and maintained as a controlled vocabulary external to this document at the URL listed above. The URI for each term is built as <BaseURL>/<Vocabulary>/<Term>, where Term is one of:

### EQUATORIAL Frames

**ICRS** : International Celestial Reference System

**FK4** : Fundamental Katalog, system 4; Besselian.
Requires Equinox; default B1950.0

**FK5** : Fundamental Katalog, system 5; Julian.
Requires Equinox; default J2000.0

### ECLIPTIC Frames

**ECLIPTIC** : Ecliptic coordinates

### GALACTIC Frames

**GALACTIC_I** : Old Galactic coordinates

**GALACTIC_II** : "New" Galactic coordinates

**SUPER_GALACTIC** : Super-galactic coordinates
pole at GALACTIC_II (47.37, +6.32); origin at GALACTIC_II (137.37, 0)

### GEOGRAPHIC Frames

**GEO_C** : Geographic (geocentric) coordinates: longitude, latitude, geocentric distance

**GEO_D** : Geodetic coordinates: longitude, latitude, elevation
Semi-major axis and inverse flattening of the reference spheroid may need to be provided; default is IAU 1976 (6378140 m, 298.2577)

**MAG** : Geomagnetic coordinates
See (Fränz and Harper, 2002)

**GSE** : Geocentric Solar Ecliptic coordinates
See (Fränz and Harper, 2002)

**GSM** : Geocentric Solar Magnetic coordinates
See (Fränz and Harper, 2002)

## SOLAR Frames

**SM** : Solar Magnetic coordinates
See (Fränz and Harper, 2002)

**HGC** : Heliographic coordinates (Carrington)
See (Seidelmann, 1992), Section 7.2; (Thompson, 2006), Section 2.2

**HGS** : Heliographic coordinates (Stonyhurst)
See (Seidelmann, 1992), Section 7.2; (Thompson, 2006), Section 2.2

**HEEQ** : Heliographic Earth Equatorial coordinates
See (Fränz and Harper, 2002) related to Heliographic (Stonyhurst); (Thompson, 2006), Section 2.2

**HRTN** : Heliographic Radial-Tangential-Normal coordinates
See (Fränz and Harper, 2002)

**HPC** : Helioprojective Cartesian coordinates
See (Thompson, 2006), Section 4.1, 2- or 3-dimensionsl (angular coordinates); left handed.

**HPR** : Helioprojective Polar coordinates
See (Thompson, 2006), Section 4.1, 2-dimensionsl (angular coordinates); left handed.

**HCC** : Heliocentric Cartesian coordinates
See (Thompson, 2006), Section 3.1, (linear coordinates); right handed.

**HGI** : Heliographic Inertial coordinates
See (Fränz and Harper, 2002)

## PLANETARY Frames

**MERCURY_C** : Planetocentric coordinates on Mercury
See (Seidelmann, 1992), Section 7.4

**VENUS_C** : Planetocentric coordinates on Venus
See (Seidelmann, 1992), Section 7.4

**LUNA_C** : Selenocentric coordinates
See (Seidelmann, 1992), Section 7.3

**MARS_C** : Planetocentric coordinates on Mars
See (Seidelmann, 1992), Section 7.4

**JUPITER_C_III** : Planetocentric coordinates on Jupiter, system III
See (Seidelmann, 1992), Section 7.4

**SATURN_C_III** : Planetocentric coordinates on Saturn, system III
See (Seidelmann, 1992), Section 7.4

**URANUS_C_III** : Planetocentric coordinates on Uranus, system III
See (Seidelmann, 1992), Section 7.4

**NEPTUNE_C_III** : Planetocentric coordinates on Neptune, system III
See (Seidelmann, 1992), Section 7.4

**PLUTO_C** : Planetocentric coordinates on Pluto, system III
See (Seidelmann, 1992), Section 7.4

**MERCURY_G** : Planetographic coordinates on Mercury
See (Seidelmann, 1992), Section 7.4

**VENUS_G** : Planetographic coordinates on Venus
See (Seidelmann, 1992), Section 7.4

**LUNA_G** : Selenographic coordinates
See (Seidelmann, 1992), Section 7.3

**MARS_G** : Planetographic coordinates on Mars
See (Seidelmann, 1992), Section 7.4

**JUPITER_G_III** : Planetographic coordinates on Jupiter, system III
See (Seidelmann, 1992), Section 7.4

**SATURN_G_III** : Planetographic coordinates on Saturn, system III
See (Seidelmann, 1992), Section 7.4

**URANUS_G_III** : Planetographic coordinates on Uranus, system III
See (Seidelmann, 1992), Section 7.4

**NEPTUNE_G_III** : Planetographic coordinates on Neptune, system III
See (Seidelmann, 1992), Section 7.4

**PLUTO_G** : Planetographic coordinates on Pluto
See (Seidelmann, 1992), Section 7.4

### OTHER Frames

**AZ_EL** : Local azimuth and elevation
Ground-based observations; Azimuth from North through East

**BODY** : Generic "BODY" coordinates

**UNKNOWN** : Unknown reference frame
Only to be used as a last resort or for simulations. The client is responsible for assigning a suitable default.

## C.2 Standard Reference Position (StdRefPos)

**BaseURL**: http://ivoa.net/vocabularies/coords
**Vocabulary**: ReferenceFrames

We include here the initial Standard Reference Position Vocabulary. The formal list is stored and maintained as a controlled vocabulary external to this document at the URL listed above. The URI for each term is built as <BaseURL>/<Vocabulary>/<Term>, where Term is one of:

**TOPOCENTER** : "Local"; in most cases this will mean: The location of the telescope.

**BARYCENTER** : Center of the solar system barycenter

**HELIOCENTER** : Center of the sun.

**GEOCENTER** : Center of the Earth.

**GALACTIC_CENTER** : Center of the Galaxy: 220 km s-1 in the direction of GALACTIC_II(90.0) w.r.t. LSRD

**LOCAL_GROUP_CENTER** : Center of Local Group: 300 km s-1 in the direction of GALACTIC_II(90.0) w.r.t. BARYCENTER
Only to be used for Redshifts and Doppler Velocities.

**MOON** : Center of the Moon.

**EMBARYCENTER** : Earth-moon barycenter

**MERCURY** : Center of Mercury.

**VENUS** : Center of Venus.

**MARS** : Center of Mars.

**JUPITER** : Center of Jupiter.

**SATURN** : Center of Saturn.

**URANUS** : Center of Uranus.

**NEPTUNE** : Center of Neptune.

**PLUTO** : Center of Pluto.

**RELOCATABLE** : Relocatable center.
For simulations, only to be used for Spatial coordinates.

**LSR** : Same as LSRK.

**LSRK** : Kinematic Local Standard of Rest: 20 km s-1 in the direction of GALACTIC_II(56,+23).
Only to be used for Redshifts and Doppler Velocities.

**LSRD** : Dynamic Local Standard of Rest: 16.6 km s-1 in the direction of GALACTIC_II(53, +25).
Only to be used for Redshifts and Doppler Velocities.

**UNKNOWN** : Unknown reference position.
Only to be used as a last resort. The client is responsible for assigning a suitable default.

## C.3 Standard Time Scale (TimeScale)

**BaseURL**: http://ivoa.net/vocabularies/coords
**Vocabulary**: TimeScale

We include here the initial Standard Time Scale Vocabulary. The formal list is stored and maintained as a controlled vocabulary external to this document at the URL listed above. The URI for each term is built as <BaseURL>/<Vocabulary>/<Term>, where Term is one of:

**TT** : Terrestrial Time

**TDT** : Terrestrial Dynamic Time; synonym for TT.

**ET** : Ephemeris Time: predecessor of, and continuous with, TT

**TAI** : International Atomic Time; 32.184 s behind TT.

**IAT** : Synonym for TAI

**UTC** : Coordinated Universal Time; 32 s behind TAI in 2000-2005.
Includes leap seconds. Pre-1972 times will be assumed to be UT/GMT.

**GPS** : Global Positioning System time scale; 19 s behind TAI, 51.184 s behind TT.

**TDB** : Barycentric Dynamical Time; synchronous with TT, except for variations in Earth ortial motion.
Requires specification of the solar system and planetary ephemeris used.

**TEB** : Barycentric Ephemeris Time; independent variable in solar system ephemeris, linear function of TT.
In most cases where TEB is specified, TDB is really the one used.

**TCG** : Geocentric Coordinate Time; properly relativistic time, running a factor 7*10^-10 faster than TT

**TCB** : Barycentric Coordinate Time; properly relativistic time, running a factor of 1.5*10^-8 faster than TDB.

**LST** : Local Siderial Time
Ground based observations only.

**LOCAL** : 'Local' time
Only to be used for simulations, in conjunction with RELOCATABLE spatial coordinates.

# D Changes from Previous Versions

No previous versions yet.

# E Modeling Conventions

This model follows the VO-DML modeling practices, however, the UML representations may vary depending on the tool used. Below we describe the graphical representation of the modeling concepts and relations.

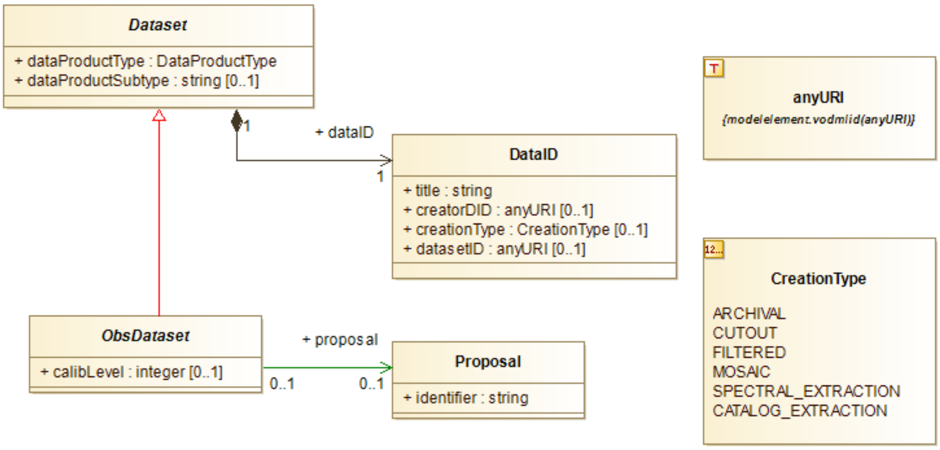



*Figure 10:* Notation example diagram

## E.1 Class

Classes are represented by a plain box. The class name is annotated in the top window, abstract classes use italic typeface. Attributes, if any, are listed in the lower panel. Attributes may only be of primitive type (real, string, etc), a defined DataType, or an Enumeration type. Relationships to other objects are defined via the composition and reference relation arrows.

## E.2 DataType

DataTypes are represented by a box shape similar to Class, but annotated with a "T" symbol in the top left corner.

## E.3 Enumerations

Enumerations are represented by a box shape similar to Class, but annotated with a "1,2.." symbol in the top left corner. Enumeration Literals (possible values) are listed below the enumeration class name.

## E.4 Generalization

Generalizations are represented by a red line, with open triangle at the end of the source, or more general, object.

## E.5 Composition

The composition relation is indicated by a black line with a solid diamond attached to the containing object, and an arrow pointing to the object being contained. The composition relation is very tight, where the container is responsible for the creation and existence of the target. Any object may be in no more than one composition relation with any container. The attribute name for the composition relation is annotated at the destination of the relation (e.g. "+ dataID"). This is typically a lower-cased version of the destination class name, but this is not required.

## E.6 Reference

The reference relation is indicated by a green line, with an arrow pointing to the object being referenced. The reference relation is much looser than composition, the container has no ownership of the target, but merely holds a pointer, or other indirect connection to it. The attribute name is annotated at the destination of the relation ( e.g. "+ proposal"). This is typically a lower-cased version of the destination class name, but may be another name indicating the role that the class is playing in this context.

## E.7 Multiplicity

All attributes and relations have a multiplicity associated with them. For attributes, the multiplicity is contained within brackets just after the attribute name. If no bracket is displayed, this is equivalent to '[1]'.

- 1 = one and only one value must be provided.
- 0..1 = zero or one value may be provided.
- * = zero or more values may be provided (open ended).

# F Data Types

## F.1 Base Data Types

Provides a set of standardized primitive data types as well as types for representing quantities ( values with associated units ). We provide a diagram of the model here, and refer the reader to Section 5 of the VO-DML modeling specification document (Lemson and Laurino et al., 2018) for more information.

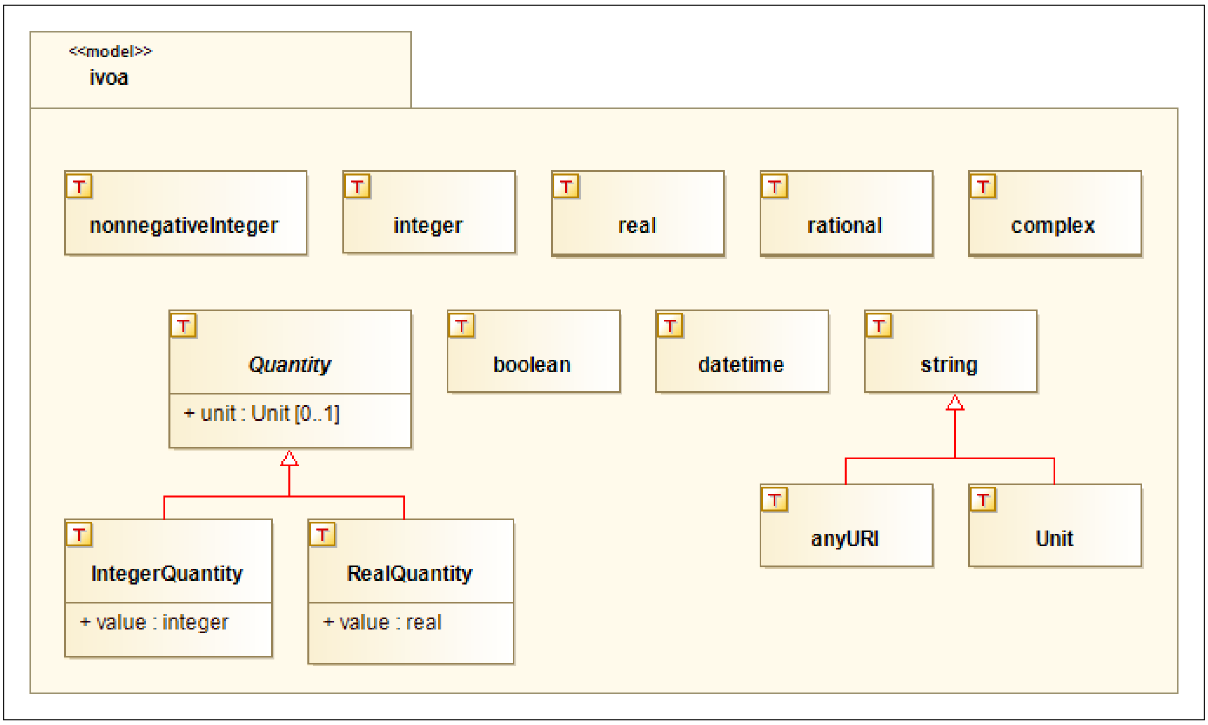



*Figure 11:* Base Data Types

### F.1.1 Units

This model requires the use of the IVOA VOUnits Standard (Demleitner and Derriere et al., 2014) for representing units of physical quantities. This standard reconciles common practices and current standards for use within the IVOA community.

### F.1.2 Dates

The 'datetime' datatype is for expressing date-time values. The string representation of a date-time value should follow the FITS convention for representing dates. The FITS standard is effectively ISO8601 format without the "Z" tag to indicate UTC (YYYY-MM-DDThh:mm:ss). Values are nominally expressed in UTC.

## References


Arviset, C., Gaudet, S. and the IVOA Technical Coordination Group (2010), 'IVOA architecture', IVOA Note.
http://www.ivoa.net/documents/Notes/IVOAArchitecture

Bradner, S. (1997), 'Key words for use in RFCs to indicate requirement levels', RFC 2119.
http://www.ietf.org/rfc/rfc2119.txt

Demleitner, M., Derriere, S., Gray, N., Louys, M. and Ochsenbein, F. (2014), 'Units in the VO, version 1.0', IVOA Recommendation.
http://www.ivoa.net/documents/VOUnits/index.html

Fränz, M. and Harper, D. (2002), 'Heliospheric coordinate systems', Planetary and Space Science, Vol 50.
http://adsabs.harvard.edu/abs/2002P&SS...50..217F

Hanisch, R. J., Farris, A., Greisen, E. W., Pence, W. D., Schlesinger, B. M., Teuben, P. J., Thompson, R. W. and Warnock, III, A. (2001), 'Definition of the Flexible Image Transport System (FITS)'.
http://adsabs.harvard.edu/abs/2001A%26A...376..359H

Lemson, G., Laurino, O., Bourges, L., Cresitello-Dittmar, M., Demleitner, M., Donaldson, T., Dowler, P., Graham, M., Gray, N., Michel, L. and Salgado, J. (2018), 'Vodml: a consistent modeling language for ivoa data models, version 1.0', IVOA Recommendation.
http://www.ivoa.net/documents/VODML/index.html

Rots, A. (2007), 'Space-time coordinate metadata for the virtual observatory', IVOA Recommendation.
http://www.ivoa.net/documents/latest/STC.html

Rots, A. H., Bunclark, P. S., Calabretta, M. R., Allen, S. L., Manchester, R. N. and Thompson, W. T. (2015), 'Representations of time coordinates in FITS. Time and relative dimension in space', *Astronomy and Astrophysics* **574**, A36, arXiv:1409.7583.
http://ads.ari.uni-heidelberg.de/abs/2015A%26A...574A..36R

Seidelmann, E. P. K. (1992), 'Explanatory supplement to the astronomical almanac', University Science Books.
http://adsabs.harvard.edu/abs/1992esta.book.....S

Thompson, W. T. (2006), 'Coordinate systems for solar image data', Astronomy and Astrophysics, Vol 449 .
http://adsabs.harvard.edu/abs/2006A&A...449..791T