Astronomical Coordinates and Coordinate Systems

Version 1.0

IVOA Working Draft 2018-11-30

Working group
Data Model Working Group
This version
http://www.ivoa.net/documents/Coordinates/20181130
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

This is an IVOA Working Draft for review by IVOA members and other interested parties. It is a draft document and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use IVOA Working Drafts as reference materials or to cite them as other than "work in progress".

A list of current IVOA Recommendations and other technical documents can be found at http://www.ivoa.net/documents/.

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, satisfying the requirements.

1.2 Requirements

The primary goal of this document is the specification of a Data Model for coordinate metadata that satisfies the following requirements; the Data Model SHALL:

1. Cover all coordinate axes of observable space: Time, Space, Electromagnetic Spectrum, Redshift (or Doppler Velocity), Polarization, and have the ability to cover any other incidental coordinates (e.g., temperature)

2. Provide metadata that are complete

3. Provide metadata that are unambiguous
4. Provide metadata that are accurate
5. Conforms to VO-DML (Lemson and Laurino et al., 2018) modeling practices.
6. Allow usage of only relevant subsets of the metadata, with the proviso that they MUST satisfy Requirements 2, 3, and 4
7. Be extensible

1.3 Context and Scope

This document is a result 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 has allowed 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 basic model for constructing coordinate frames and for collecting them in coordinate systems
- the general model for specifying coordinate values
- simple, concrete, domain specific coordinate types for the most common use cases.

1.4 Role within the VO Architecture

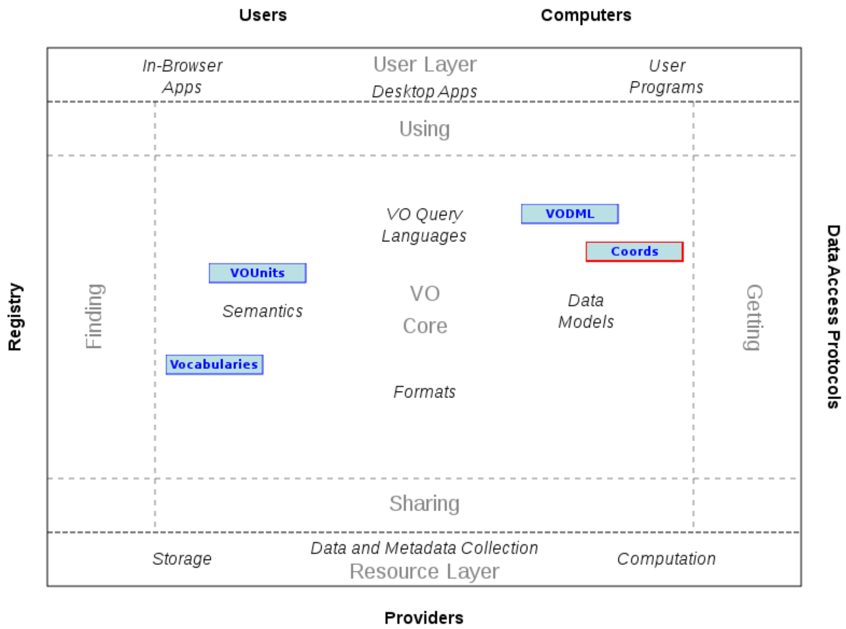

Figure 1: Architecture diagram for this document

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

2 Model: coords

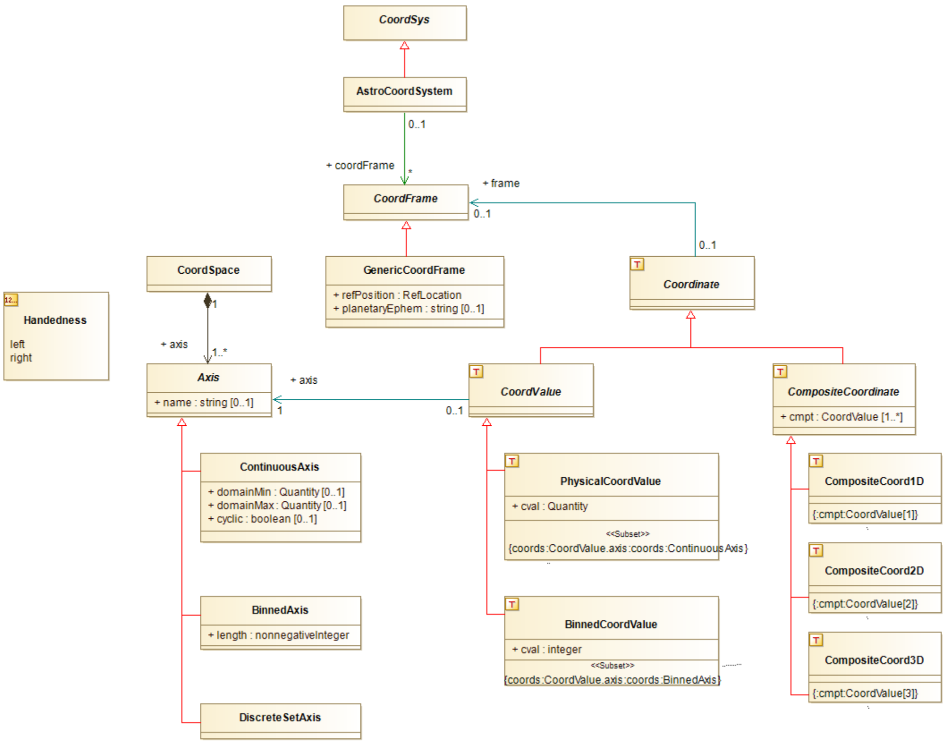

Figure 2: Base elements of Coords model

The base 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.

2.1 CoordSpace

This object defines a domain space. ie: it describes the set of possible coordinate values. For many cases, users will not need to define coordinate spaces explicitely, but instead refer to pre-defined descriptions for commonly used types (eg: CARTESIAN, SPHERICAL, CYLINDRICAL).

2.1.1 CoordSpace.axis

vodml-id: CoordSpace.axis
type: coords:Axis

multiplicity: 1..*
Describes an axis of the coordinate space.

2.2 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 for extension for other types as needed.

2.2.1 Axis.name

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

2.3 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.

2.3.1 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).

2.3.2 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).

2.3.3 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.

2.4 BinnedAxis

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

2.4.1 BinnedAxis.length

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

2.5 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.

2.6 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.

2.7 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.

2.8 GenericCoordFrame

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

2.8.1 GenericCoordFrame.refPosition

vodml-id: GenericCoordFrame.refPosition
type: coords:domain.space.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.

2.8.2 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"

2.9 Coordinate (Abstract)

Abstract base class for the Coordinate data types. All Coordinates MAY refer to a coordinate frame, providing additional metadata relevant to interpreting the coordinate value.

2.9.1 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.

2.10 CoordValue (Abstract)

Abstract head of the 1-dimensional 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: CoordValue and its children provide a generic means for describing any sort of data. This model also provides a set of simple, specialized Coordinate types for the most common data which encapsulate much of this content into the definition of the class itself. These can be found in the relevant domain (Space, Time, etc) packages.

2.10.1 CoordValue.axis

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

2.11 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:CoordValue.axis
  type: coords:ContinuousAxis

2.11.1 PhysicalCoordValue.cval

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

2.12 BinnedCoordValue

Coordinate value type specifically intended for binned data (eg: pixel indexes).
subset
  role: coords:CoordValue.axis
  type: coords:BinnedAxis

2.12.1 BinnedCoordValue.cval

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

2.13 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.

2.13.1 CompositeCoordinate.cmpt

vodml-id: CompositeCoordinate.cmpt
type: coords:CoordValue
multiplicity: 1..*
Component member of the composite coordinate.

2.14 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]

2.15 CompositeCoord2D

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

2.16 CompositeCoord3D

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

2.17 CoordSys (Abstract)

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

2.18 AstroCoordSystem

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

2.18.1 AstroCoordSystem.coordFrame

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

3 Package: domain

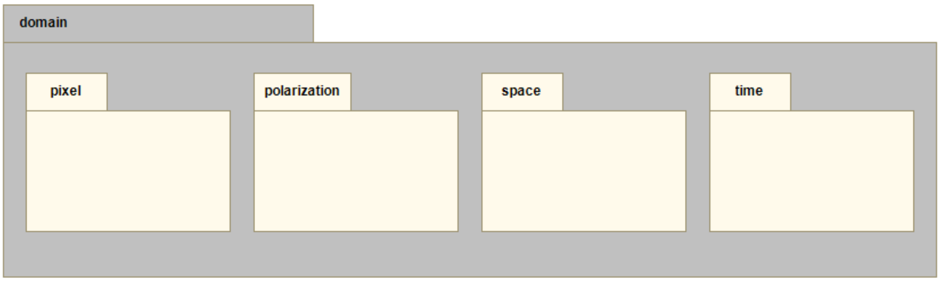

Figure 3: Domain package overview

The Domain package contains content specialized for certain physical domains (Space, Time, etc). The content provided here is considered generally useful in a wide range of cases. Other domains may be added here, or retained with the model which utilizes that content (eg: Photometric types may be added here, or be included in the Spectral or Photometry model).

4 Package: domain.pixel

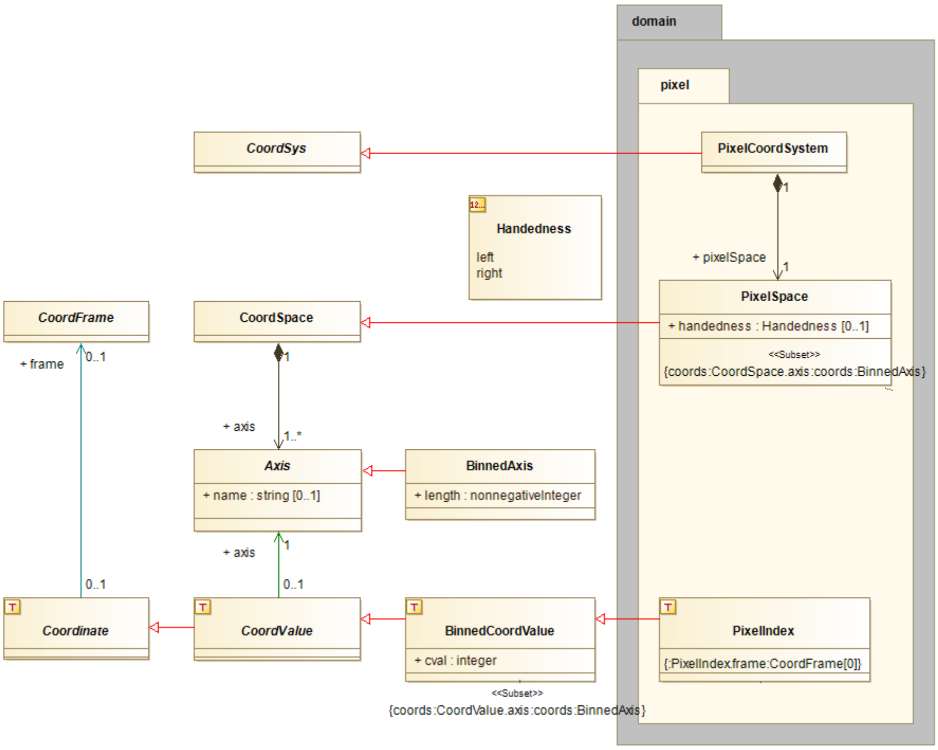

Figure 4: Pixel domain model elements

This package provides 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 indexes into that space.

4.1 PixelCoordSystem

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

4.1.1 PixelCoordSystem.pixelSpace

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

4.2 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

4.2.1 PixelSpace.handedness

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

4.3 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]

5 Package: domain.space

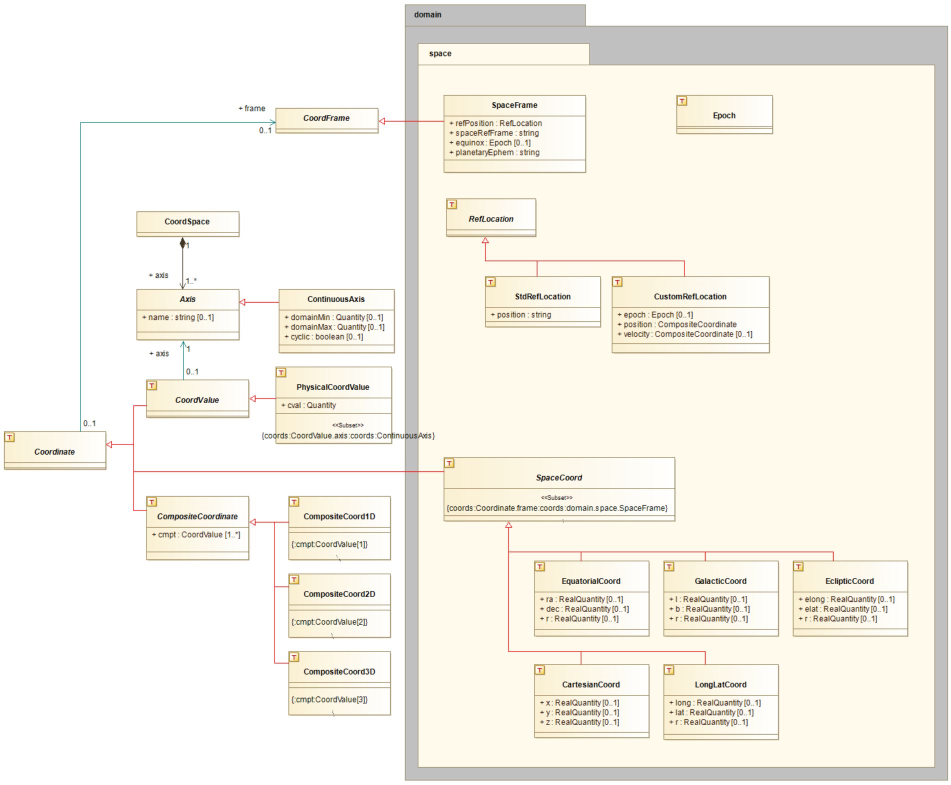

Figure 5: Space domain model elements

This package provides specialized content for the Space domain. The spatial domain requires a Space Frame, giving the reference frame (orientation) and reference position (origin). We also define here a set of simplified spatial coordinate types for the most common cases.

5.1 SpaceFrame

A Space Frame is specified by its Reference Frame (currently only standard reference frames are allowed), and a Reference Position. An equinox MUST be provided for pre-ICRS reference frames. A planetary ephemeris MAY be provided if relevant; if not provided, it is assumed to be "DE 405".

5.1.1 SpaceFrame.refPosition

vodml-id: domain.space.SpaceFrame.refPosition
type: coords:domain.space.RefLocation

multiplicity: 1
The spatial location at which the coordinates are considered to have been determined. We support both standard and custom reference positions.

5.1.2 SpaceFrame.spaceRefFrame

vodml-id: domain.space.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.

5.1.3 SpaceFrame.equinox

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

5.1.4 SpaceFrame.planetaryEphem

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

5.2 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. eg: "B1950", "J2000.0"

5.3 RefLocation (Abstract)

RefLocation defines the origin of the spatial coordinate space. This location is represented either by a standard reference position (which 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.

5.4 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 red 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.

5.4.1 StdRefLocation.position

vodml-id: domain.space.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.

5.5 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.

5.5.1 CustomRefLocation.epoch

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

5.5.2 CustomRefLocation.position

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

5.5.3 CustomRefLocation.velocity

vodml-id: domain.space.CustomRefLocation.velocity
type: coords:CompositeCoordinate

multiplicity: 0..1
The velocity of the reference location.

5.6 SpaceCoord (Abstract)

Abstract head of a set of specialized spatial coordinate types which cover the most commonly used cases. The definitions of these 'shortcut' types include a complete description of the associated coordinate space in which they reside, thereby reducing the need to include 'boilerplate' content. It is expected that these coordinates will be used in the vast majority of cases.
subset
  role: coords:Coordinate.frame
  type: coords:domain.space.SpaceFrame

5.7 CartesianCoord

A spatial location in a standard 3D Cartesian coordinate space (see: A.1). All axes are optional, users need only supply values for relevant axes.

5.7.1 CartesianCoord.x

vodml-id: domain.space.CartesianCoord.x
type: ivoa:RealQuantity
multiplicity: 0..1
Coordinate value along the 'X' axis.

5.7.2 CartesianCoord.y

vodml-id: domain.space.CartesianCoord.y
type: ivoa:RealQuantity
multiplicity: 0..1
Coordinate value along the 'Y' axis.

5.7.3 CartesianCoord.z

vodml-id: domain.space.CartesianCoord.z
type: ivoa:RealQuantity
multiplicity: 0..1
Coordinate value along the 'Z' axis.

5.8 EclipticCoord

A spatial location in a spherical Ecliptic coordinate space (see: A.2). The associated SpaceFrame MUST represent an Ecliptic frame.

5.8.1 EclipticCoord.elong

vodml-id: domain.space.EclipticCoord.elong
type: ivoa:RealQuantity
multiplicity: 0..1
Value along the Longitude axis.

5.8.2 EclipticCoord.elat

vodml-id: domain.space.EclipticCoord.elat
type: ivoa:RealQuantity
multiplicity: 0..1
Value along the Latitude axis.

5.8.3 EclipticCoord.r

vodml-id: domain.space.EclipticCoord.r
type: ivoa:RealQuantity
multiplicity: 0..1
Value along the Radial axis.

5.9 EquatorialCoord

A spatial location in a spherical Equatorial coordinate space (see: A.2). The associated SpaceFrame MUST represent an Equatorial frame.

5.9.1 EquatorialCoord.ra

vodml-id: domain.space.EquatorialCoord.ra
type: ivoa:RealQuantity
multiplicity: 0..1
Value along the Longitude axis, (Right Ascension)

5.9.2 EquatorialCoord.dec

vodml-id: domain.space.EquatorialCoord.dec
type: ivoa:RealQuantity
multiplicity: 0..1
Value along the Latitude axis, (Declination)

5.9.3 EquatorialCoord.r

vodml-id: domain.space.EquatorialCoord.r
type: ivoa:RealQuantity
multiplicity: 0..1
Value along the radial axis, (Distance)

5.10 GalacticCoord

A spatial location in a spherical Galactic coordinate space (see: A.2). The associated SpaceFrame MUST represent a Galactic frame.

5.10.1 GalacticCoord.l

vodml-id: domain.space.GalacticCoord.l
type: ivoa:RealQuantity
multiplicity: 0..1
Value along the Longitude axis.

5.10.2 GalacticCoord.b

vodml-id: domain.space.GalacticCoord.b
type: ivoa:RealQuantity
multiplicity: 0..1
Value along the Latitude axis.

5.10.3 GalacticCoord.r

vodml-id: domain.space.GalacticCoord.r
type: ivoa:RealQuantity
multiplicity: 0..1
Value along the Radial axis.

5.11 LongLatCoord

A spatial location in a Generic Spherical coordinate space (see: A.2).

5.11.1 LongLatCoord.long

vodml-id: domain.space.LongLatCoord.long
type: ivoa:RealQuantity
multiplicity: 0..1
Value along the Longitude axis.

5.11.2 LongLatCoord.lat

vodml-id: domain.space.LongLatCoord.lat
type: ivoa:RealQuantity
multiplicity: 0..1
Value along the Latitude axis.

5.11.3 LongLatCoord.r

vodml-id: domain.space.LongLatCoord.r
type: ivoa:RealQuantity
multiplicity: 0..1
Value along the Radial axis.

6 Package: domain.time

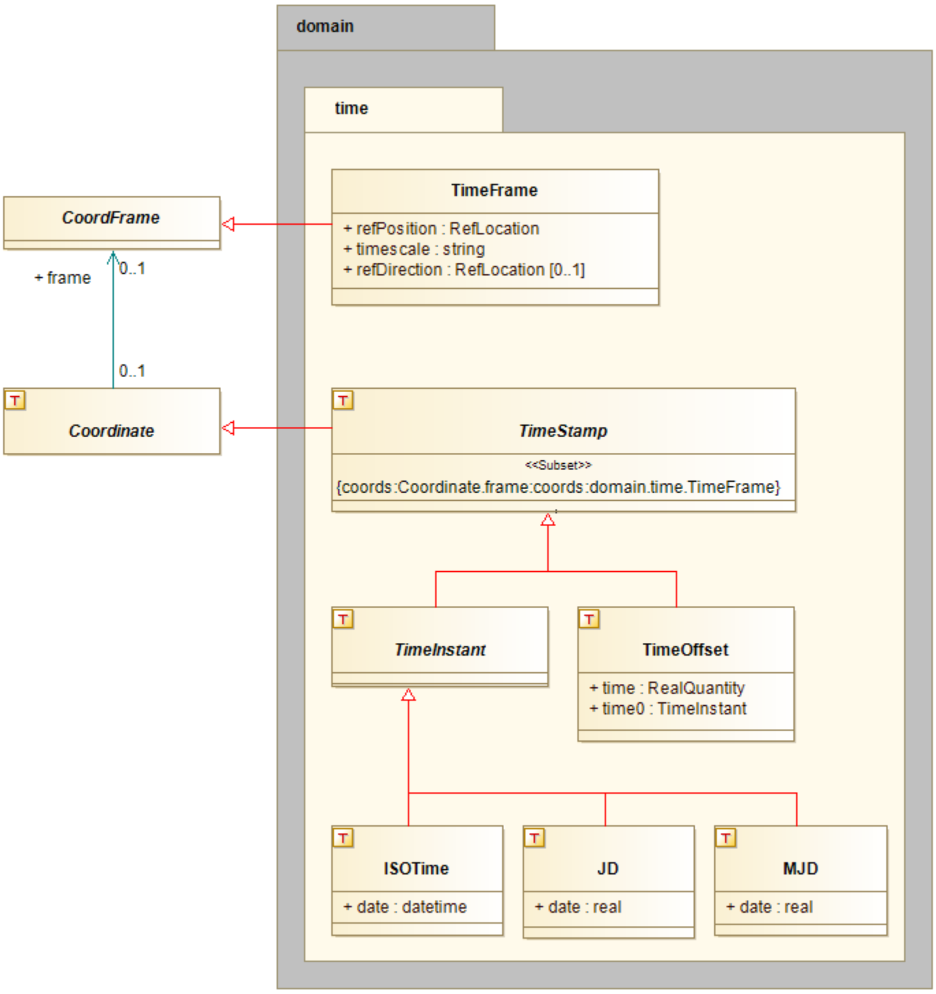

Figure 6: Time domain model elements

This package provides specialized content for the Time domain.
A Brief Primer on Time Metadata; for reference and more information, see: FITS WCS Paper IV (Rots et al., 2015 A&A 574, 36).

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 (eg: 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...]]]]. No time zone characters
   * TDB runs on average synchronously with TT, but corrects for the relativistic effects caused by deviations in the oribt 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.
   * Beware 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. Do 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. Eventually these could come into more common use, but at least for now, I assume we will be sticking with TT and TDB. More in the cited A&A paper.

6.1 TimeFrame

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

6.1.1 TimeFrame.refPosition

vodml-id: domain.time.TimeFrame.refPosition
type: coords:domain.space.RefLocation
multiplicity: 1
The spatial location at which the coordinate is considered to have been taken from.

6.1.2 TimeFrame.timescale

vodml-id: domain.time.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.

6.1.3 TimeFrame.refDirection

vodml-id: domain.time.TimeFrame.refDirection
type: coords:domain.space.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.

6.2 TimeStamp (Abstract)

This is the abstract basis for a set of simple time domain coordinates which is 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:domain.time.TimeFrame

6.3 TimeInstant (Abstract)

TimeStamps which specify a specific instant in time.

6.4 ISOTime

An instant in time expressed with structure in representation defined by the ISO-8601 standard within the restrictions imposed by the IVOA.

6.4.1 ISOTime.date

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

6.5 JD

A time stamp 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.

6.5.1 JD.date

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

6.6 MJD

A time stamp expressed in Modified Julian Days. T(MJD) = T(JD) - 2440000.5.

6.6.1 MJD.date

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

6.7 TimeOffset

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

6.7.1 TimeOffset.time

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

6.7.2 TimeOffset.time0

vodml-id: domain.time.TimeOffset.time0
type: coords:domain.time.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 Package: domain.polarization

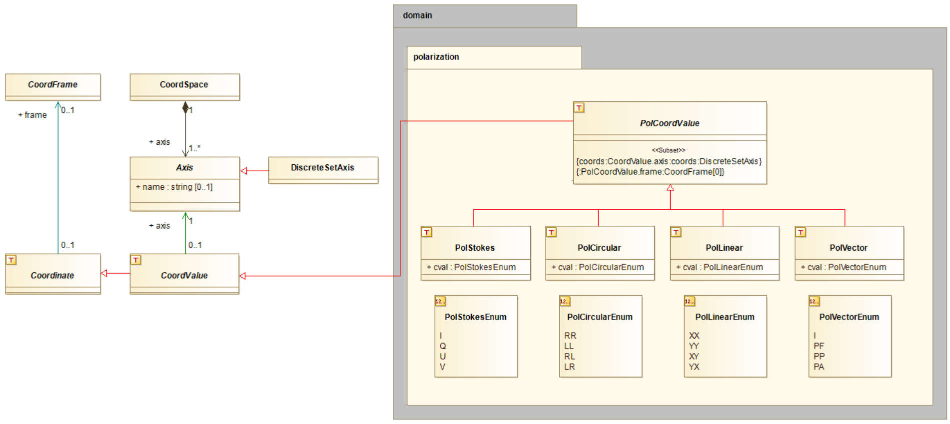

Figure 7: Polarization domain model elements

This package provides specialized content for the Polarization domain. This domain provides a concrete example of a Discrete coordinate space. The coordinates in this domain are given by enumerated lists of the various polarization types.

7.1 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:CoordValue.axis
  type: coords:DiscreteSetAxis

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

7.2 PolStokes

Coordinate for Stokes Polarization type

7.2.1 PolStokes.cval

vodml-id: domain.polarization.PolStokes.cval
type: coords:domain.polarization.PolStokesEnum

multiplicity: 1
The coordinate value MUST be from the PolStokesEnum enumerated set.

7.3 PolCircular

Coordinate for Circular Polarization type

7.3.1 PolCircular.cval

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

7.4 PolLinear

Coordinate for LinearPolarization type

7.4.1 PolLinear.cval

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

7.5 PolVector

Coordinate for Vector Polarization type

7.5.1 PolVector.cval

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

7.6 PolStokesEnum

Stokes Polarization states
<u>Enumeration Literals</u>

- **I** : **vodml-id:** domain.polarization.PolStokesEnum.I
- **Q** : **vodml-id:** domain.polarization.PolStokesEnum.Q
- **U** : **vodml-id:** domain.polarization.PolStokesEnum.U
- **V** : **vodml-id:** domain.polarization.PolStokesEnum.V

7.7 PolCircularEnum

Circular Polarization states

Enumeration Literals

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

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

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

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

7.8 PolLinearEnum

Linear Polarization states

Enumeration Literals

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

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

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

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

7.9 PolVectorEnum

Vector Polarization states

Enumeration Literals

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

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

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

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

A 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.

A.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

A.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

A.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

B Standard Vocabularies

B.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.

B.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.

B.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.

C Changes from Previous Versions

No previous versions yet.

D 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.

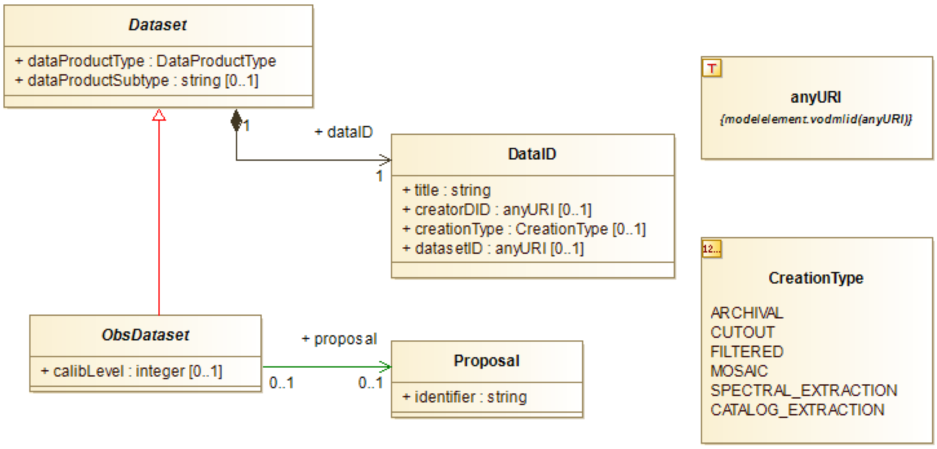

Figure 8: Notation example diagram

D.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.

D.2 DataType

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

D.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.

D.4 Generalization

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

D.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.

D.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.

D.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).

E Data Types

E.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.

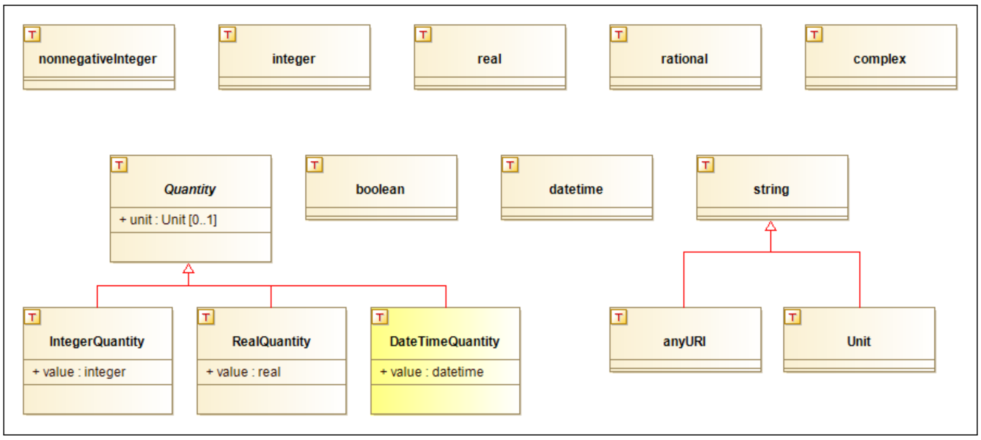

Figure 9: Base Data Types

E.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.

E.1.2 UCDs

This model requires any ucd field to comply with syntax defined in ”An IVOA Standard for Unified Content Descriptors” (Derriere and Preite Martinez et al., 2005).

E.1.3 Dates

The ’datetime’ datatype is for expressing date-time values. The string representation of a datetime 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

Derriere, S., Preite Martinez, A., Williams, R., Gray, N., Mann, R., McDowell, J., Mc Glynn, T., Ochsenbein, F., Osuna, P. and Rixon, G. (2005), 'An IVOA Standard for Unified Content Descriptors Version 1.10', IVOA Recommendation 19 August 2005, arXiv:1110.0525.
http://adsabs.harvard.edu/abs/2005ivoa.spec.0819D

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

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

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