International
Virtual
Observatory
Alliance

IVOA Dataset Metadata Model

Version 1.0

IVOA Working Draft 20140930

This version:

WD-DatasetDM-1.0-20140930

Previous version(s):

Editor(s):

Mark Cresitello-Dittmar

Authors:

Francois Bonnarel, Omar Laurino, Gerard Lemson, Mireille Louys, Arnold Rots, Doug Tody, and the IVOA Data Model Working Group.

Abstract

This document provides a data model describing the structure and content of generic Dataset metadata for the IVOA. This is a high-level model which is to be referenced and extended by other models describing specific types of Datasets and Data products. In this document, we specify the generic Dataset, as well as an ObservationDataset model which covers the class of Datasets which are derived from an Observation. At the time of this writing, there is no formal Observation-Experiment model for the IVOA, so we include a hypothetical Observation-Experiment model to serve as a placeholder.

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".

Acknowledgements

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 European Commission's Sixth Framework Program http://fp6.cordis.lu/fp6/home.cfm via the Optical Infrared Coordination Network (OPTICON), http://www.astro-opticon.org.

Change Log:

2014 Jun 13: Initial draft of version 1.0
2014 Sep 30: Draft revised with STC2 prototype, initial draft feedback, and updates due to Cube model development.

Contents

1 Introduction

1.1 Motivation

All IVOA datasets must contain a common set of metadata elements to facilitate the registration, discovery, and interoperability of these datasets. To date, individual IVOA data models have independently defined this metadata within the separate documents. This has resulted in some level of inconsistency between models, as well as document bloat, and some ambiguity as to the hierarchy and relation of models to each other. For example, the ObsCore-1.0 model describes itself as defining "the core components of the Observation data model ", but there is no formal definition of an Observation data model in the IVOA. Without this higher-level document, it is difficult for detailed models to properly reference and/or extend this content consistently.

With the development of the Cube model, significant effort has been made to properly model this high-level metadata, and separate the components related to the generic dataset, a dataset derived from an observation, and the observation itself. This document represents the results of that effort. Here, we define the generic dataset metadata, and provide an example for extending this with metadata related to datasets resulting from a specific process (Observation). As such, the ObsCore model should be considered a 'view' of this model, highlighting the core components required for supporting TAP services.

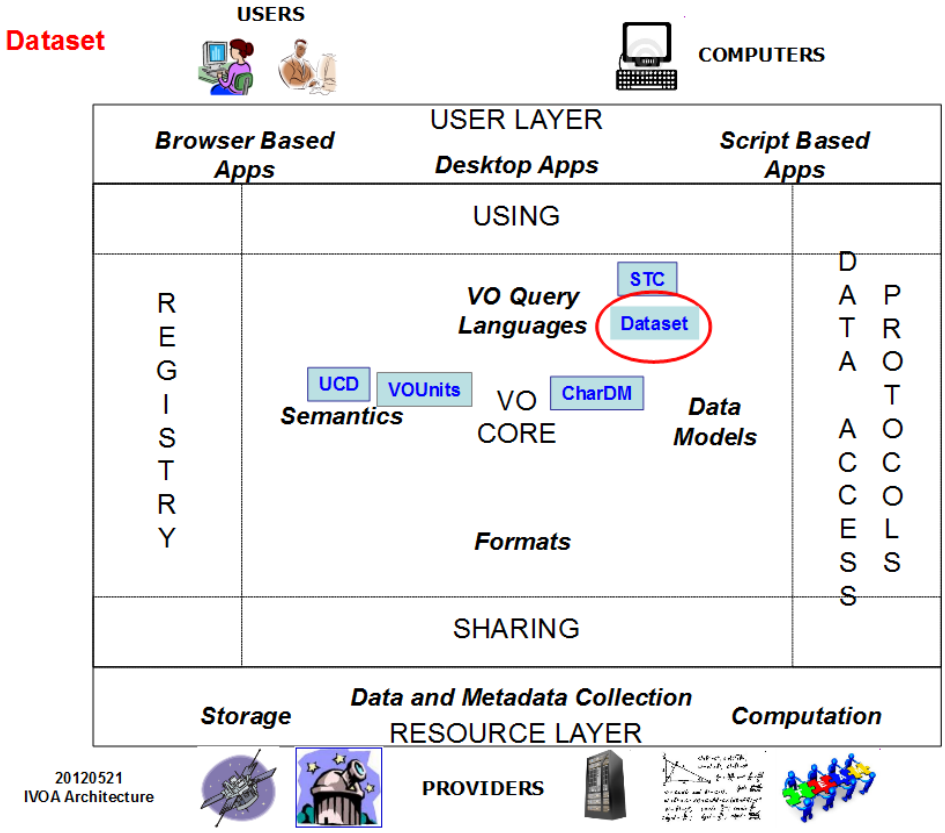
The descriptions of many elements of this model are a result of reviewing and combining those contained in the ObsCore (1.0), Spectral (2.0), and Characterisation (1.13) models. As such, it represents a uniform, consistent description set. Furture revisions of those documents should be defined with respect to this model.

1.2 Requirements

The primary goals of this document are:

- to provide a specification of generic dataset metadata.
- to specify metadata associated with an Observation (experiment) which are to be included in datasets derived from observations (ObsDataset).

1.3 IVOA Architecture Context



1.4 Model Dependencies

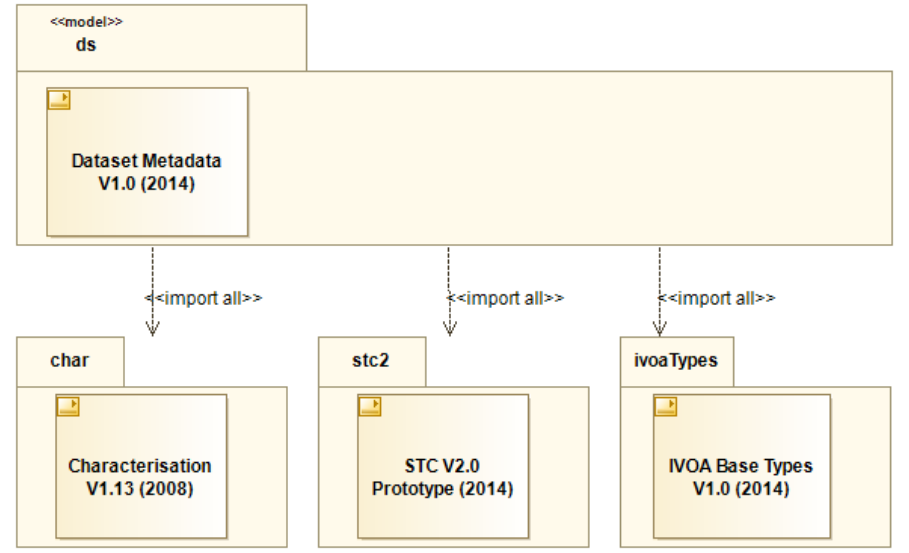


Figure 1: Model dependency diagram

The Dataset model is built on other data models as indicated in Figure 1. The <<model>> and <<import>> stereotypes provide information identifying the model, it's version, any dependencies, and URLs to find more information about the model definitions including HTML and schema documentation. See Appendix B for more information about the content of these stereotypes and how they are used in serializations.

1.5 Structure of this Documentation

+ Major sections for each model area (Dataset, Observation, etc.).
+ First subsection in each section is the primary element within that model
+ Subsequent subsections for secondary elements, in alphabetical order.
+ Each subsection has sub-subsections for each attribute/relation
    - attributes show the full definition including datatype and usage.
    - relations describe the usage of the object in that context, and a reference to the full definition.

2 Dataset Model

This section describes the generic, high-level metadata associated with an IVOA Dataset. Since serialization format choices may effect the number of files or components which comprise a dataset, we define an IVOA Dataset as "a file or files which are considered to be a single deliverable". Examples of viable datasets include:

+ An individual data product, such as a Spectrum, or Image.

+ A 'tar' file or directory of processed observational data files.

This metadata identifies the dataset, and provides information regarding the ownership, rights and associations with other datasets. The primary purpose of this metadata is to facilitate the registry and discovery of datasets within the IVOA community.

Several of the objects modeled here are based on descriptions given in the IVOA document, "Resource Metadata for the Virtual Observatory; Version 1.12" [1] (Resource Metadata). Where applicable, we provide the appropriate citation in the text below.

2.1 Dataset

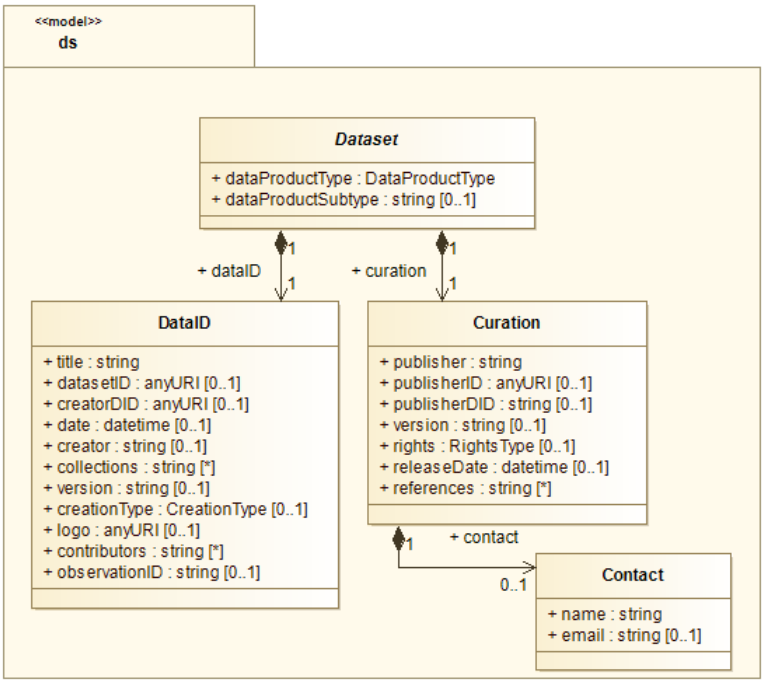


Figure 2: Generic Dataset Metadata

Abstract object for the generic IVOA Dataset. It is intended to be useful for any type of data. Specific dataset models should extend this object, providing detailed definitions and additional content as appropriate for that type of dataset.

2.1.1 Dataset.dataProductType:DataProductType

Describes the high level scientific classification of the data content. This enumerated string conveys the general idea of the content and organization of a dataset. Allowed values are provided in the DataProductType enumeration list detailed in Section 5.2.1.

2.1.2 Dataset.dataProductSubtype:string

Secondary type identifier for the dataset. This field is intended to precisely specify the scientific nature of the data product, possibly in terms relevant only to a specific archive or data collection.

2.1.3 Dataset.curation:Curation

See Section 2.3 for content definition.

Provides metadata related to the entity responsible for the curation of the dataset.

2.1.4 Dataset.dataID:DataID

See Section 2.4 for content definition.

DataID provides high level identification metadata for the dataset, and any associations with various collections.

2.2 Contact

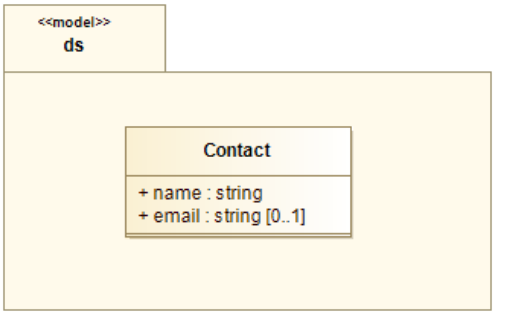


Figure 3: Contact detail diagram

Contact information for a person or entity.

2.2.1 Contact.name:string

Name of the contact or entity. (RM:Contact.Name)

2.2.2 Contact.email:string

E-mail address of the contact. (RM:Contact.Email)

2.3 Curation

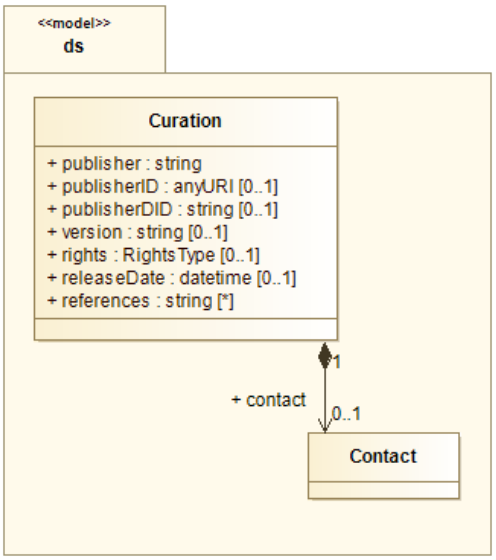


Figure 4: Curation detail diagram

The Curation object provides information about the entity responsible for the support of the dataset content. It is assembled from definitions provided by the IVOA Resource Metadata document. Here, we provide a brief description of each field for easy reference, along with a notation of its mapping to the Resource Metadata document (RM:field), where the reader may find more detailed information.

2.3.1 Curation.publisher:string

The entity making the data available. (RM:Curation.Publisher)

2.3.2 Curation.publisherID:anyURI

The identifier for the publisher, provided according to the syntax defined in "IVOA Identifiers"[2]. (RM:Curation.PublisherID)

2.3.3 Curation.publisherDID:string

This is the publisher specified dataset identifier (RM:Resource.Identifier) and may be an internal ID used by the archive. It may differ from the original identifier given by the creator of the dataset.

2.3.4 Curation.releaseDate:datetime

Date the curated dataset was last modified. (RM:Curation.Date)

NOTE: Date type and format restrictions are described in Section 5.1.3.

2.3.5 Curation.version:string

Version is provided by the publisher or creator and may be any string. (RM:Curation.Version)

2.3.6 Curation.rights:RightsType

Indicates the access privileges to the content. (RM:Collection.Rights)

RightsType enumerated value list is detailed in Section 5.2.3.

2.3.7 Curation.references:string

Set of one (1) or more bibliographic or documentation references associated with the dataset. This is a forward link to major publications which reference the dataset. (RM:General.Source)

2.3.8 Curation.contact:Contact

Contact information of the person/entity responsible for the content of the dataset. (RM:Curation.Contact) The Contact object is described in Section 2.2.

2.4 DataID

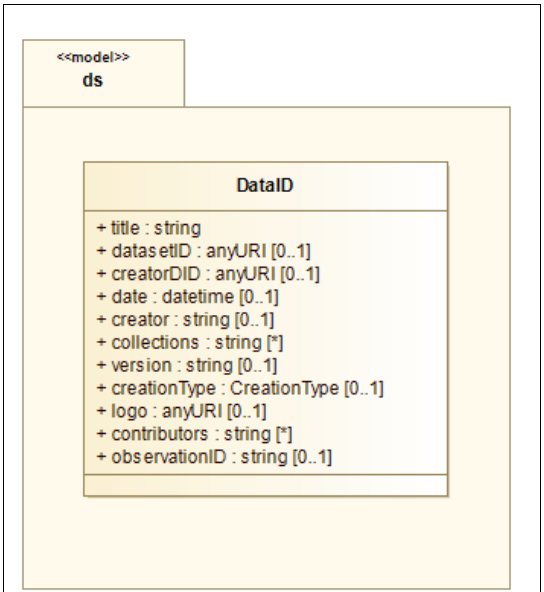


Figure 5: DataID detail diagram

The Data Identification object (DataID) stores the dataset identifiers and its membership within larger collections.

The Dataset IDs in this object must comply with the syntax for dataset identifiers defined in the "IVOA Identifiers" [2] document, including the use of 'stop' characters to identify specific datasets that are not individually in the registry.

Much of the content of this object is assembled from various definitions in the IVOA Resource Metadata document. Here, we provide a brief description of each field for easy reference, along with a notation of its mapping to the Resource Metadata document (RM:field), where the reader may find more detailed information.

2.4.1 DataID.title:string

A free form string giving a title for the dataset. (RM:Identity.Title)

2.4.2 DataID.datasetID:anyURI

For this field we recommend a journal-based URI such as the IVOA/ADEC/ADS dataset identifier. By agreement between AAS journals, the ADS and the ADEC (NASA data centers), dataset identifiers, described in http://vo.ads.harvard.edu/dv, will be used to link journal articles back to the archival datasets containing the relevant observational data. If analogous but independent systems of URI designation are later adopted by other centers (e.g. by European journals) and accepted by IVOA, they will be suitable in this field.

The value of this field may be the same as in Curation.PublisherDID.

2.4.3 DataID.creatorDID:anyURI

The Dataset ID as defined internally by the creator. It may be entirely different from the DatasetID field. It is used to identify a particular original exposure in an archive and will not necessarily change even if the VO object in question is a cutout or is otherwise further processed.

2.4.4 DataID.date:datetime

Data processing or creation date (RM:Curation.Date). See Section 5.1.3 for Date format specification.

2.4.5 DataID.creator:string

A free form string giving the name of the institution or entity which created the dataset. (RM:Curation.Creator)

2.4.6 DataID.collections:string

A free form string, generally defined and set by the creating entity. The dataset is associated with one (1) or more Collections (instrument name, survey name, etc.) indicating some degree of compatibility with other data sharing the same Collection properties. Examples: "WFC", "Sloan", "BFS Spectrograph", "MSX Galactic Plane Survey".

2.4.7 DataID.version:string

Version of the creator-produced dataset.

2.4.8 DataID.creationType:CreationType

The dataset creation type describes the nature or genre of the content. (RM:General.Type). The creationType enumeration set is defined in Section 5.2.2.

2.4.9 DataID.logo:anyURI

URL pointer to a graphical logo associated with the creator of the document content. (RM:Curation.Creator.Logo)

2.4.10 DataID.contributors:string

This field contains a set of one (1) or more strings identifying entities which contributed to the content of the document. (RM:Curation.Contributor)

2.4.11 DataID.observationID:string

Internal ID determined by the data provider to identify the observation from which the dataset was produced.

3 ObservationDataset (ObsDataset)

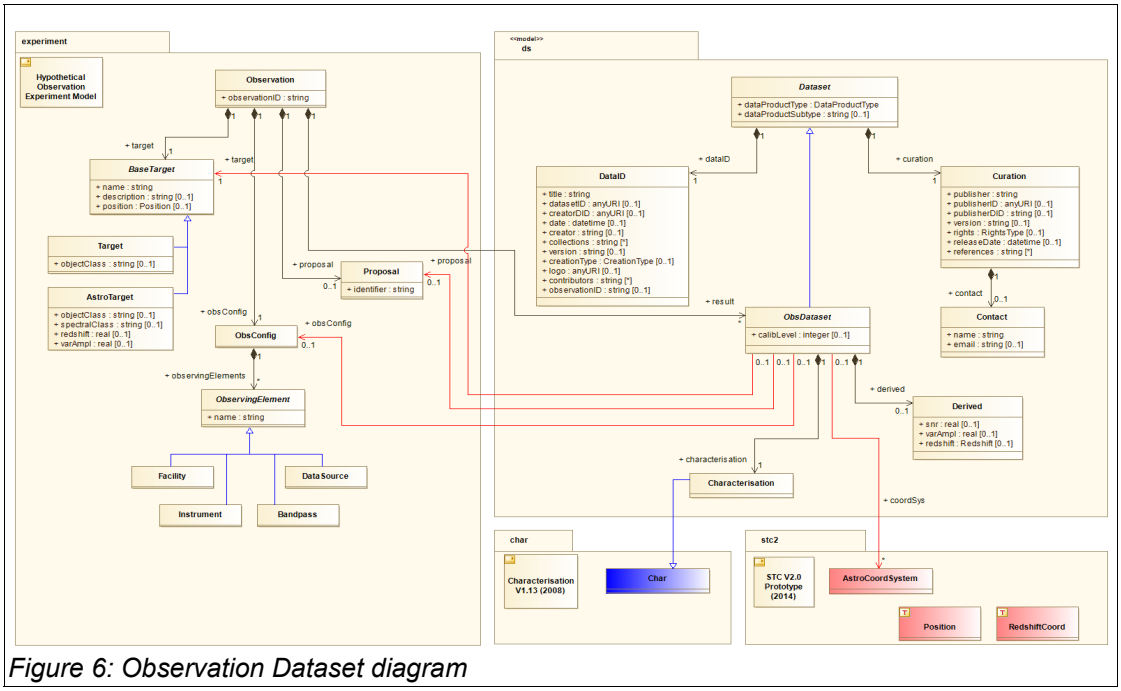


Figure 6: Observation Dataset diagram

This section defines additional metadata associated with Datasets which are derived from Observations. This metadata gives a high-level summary of the coverage of the dataset in coordinate space, as well as the coordinate systems used, and general information about the observation itself.

Many components of this object are expected to be sourced from an as yet undefined Observation model. In lieu of this model, we define a straw-man substitute in this document. The Observation model described here depicts an "Observation" as a type of "Experiment", where certain elements of the observation configuration and parameters are referenced within the ObsDataset. Other forms of "Experiment" (such as "Simulation") could fit in this framework. In fact, much of the framework is based on that shown in the Simulation data model[6].

3.1 ObsDataset

Abstract object extending Dataset with metadata relevant to datasets derived from Observations.

3.1.1 ObsDataset.calibLevel:integer

High level classification for the calibration level of a particular dataset as a whole. The calibration level concept conveys to the user information on how much data reduction/processing has been applied to the data. It is up to the data providers to consider how to map their own internal classification to the scale defined here.

Scale:

- 0 - Raw instrumental data, in a proprietary or internal data-provider defined format.

- 1 - Instrumental data in a standard format (FITS, VOTable, etc)
- 2 - Calibrated, science ready data with the instrument signature removed.
- 3 - Enhanced data products like mosaics, resampled or drizzled images, or heavily processed survey fields. Level 3 data products may represent the combination of data from multiple primary observations.

3.1.2 ObsDataset.characterisation:Characterisation

Characterisation provides a 'characteristic' view of the dataset coordinate space. For each represented domain (spatial, spectral, temporal, etc), characterisation provides metadata summarizing the coverage, resolution, representative accuracies, etc. for the dataset as a whole. See Section 3.2 for Characterisation object definition.

3.1.3 ObsDataset.coordSys:AstroCoordSystem

Zero (0) or more references to coordinate system defintions associated with the dataset. Since multiple data products may contibute to the content of a dataset, this element provides convenient, high-level access to definitions which may be distributed among lower-level objects. See Section 6.2.3 for definition of the STC2 AstroCoordSystem object.

3.1.4 ObsDataset.derived:Derived

Provides a high level summary of certain properties of the dataset. Its primary purpose is to support high level filtering of datasets during data discovery. See Section 3.3 for Derived object definition.

3.1.5 ObsDataset.obsConfig:ObsConfig

Reference to ObsConfig object from Observation. This object provides some high-level metadata related to the observation configuration. See Section 4.5 for ObsConfig object definition.

3.1.6 ObsDataset.proposal:Proposal

Reference to Proposal object from Observation. This object provides metadata identifying any proposal related to the observation which produced the dataset. See Section 4.7 for Proposal object definition.

3.1.7 ObsDataset.target:BaseTarget

Reference to a BaseTarget object from Observation. Provides metadata describing the target of the observation. See Section 4.2 for the definition of the BaseTarget object.

3.2 Characterisation

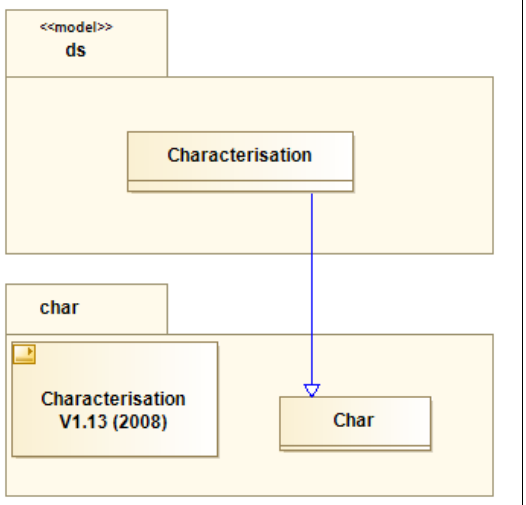


Figure 7: Characterisation Relation to Char Model

The ObsDataset Characterisation object is a direct extension of the Characterisation object from the IVOA Characterisation Data Model Standard [3]. This provides a local object which may be contained in a 'composition' relation to other objects of this model (specifically ObsDataset). This object may be extended and/or modified by specific Dataset models as needed.

Characterisation provides a broad scale description of the coordinate space occupied by the dataset in each of the axis domains. This includes axes not directly represented in the dataset 'data', for example, the Time axis of a 2D spatial image. As such, there is a strong correlation between Characterisation and Data objects. In some cases, the values may be directly derived from the data, in others, they provide supporting metadata about the broader domain. For example, the Accuracy fields in Characterisation represent typical accuracy for the dataset, while those within Data provide per-data-point errors.

3.3 Derived

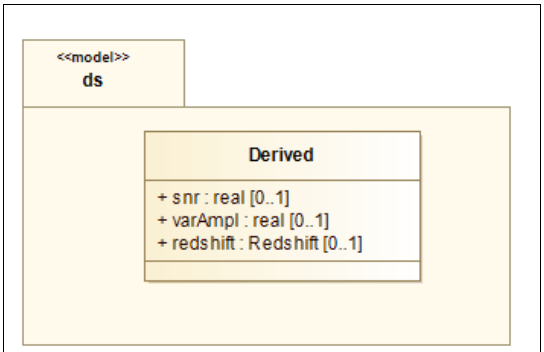


Figure 8: Derived detail diagram

The Derived (short for Derived Data) object holds summary information about the dataset obtained by evaluating or analyzing the contents of the dataset.

3.3.1 Derived.snr:real

The signal-to-noise (SNR) is provided mainly as a way for searches to exclude data whose quality is insufficient for a particular study. Data providers may use their own definition, as we do not prescribe a uniform method to calculate it. A suitable method, set forth by the STScI/STECF/CADC Spectral Container Working Group, is to define the signal as the median of the flux values in the spectrum and the noise as the median absolute third-order difference of flux values spaced two pixels apart. This noise value is then multiplied by 1.482602 / sqrt(6). A detailed description and discussion of the algorithm can be found in the ”ST-ECF newsletter; issue #42”[4]. Implementations of the algorithm can be obtained from ”stecf.org”[5].

This method describes the high-spectral-frequency noise but does not take into account intermediate-spectral-frequency background ‘noise’. Projects which are background dominated may wish to include this in the noise definition. Furthermore, most spectra vary in SNR across their waveband; users should therefore only use this single SNR as a crude selection parameter.

3.3.2 Derived.redshift:Redshift

This field represents a measurement of the redshift from the data. Redshift content is detailed in Section 5.2.4.

NOTE: There are two (2) other redshifts in our model:

+ the Target redshift: stores the Target redshift as determined by other means.

+ the SpectralFrame redshift: used only if a 'rest frame' spectrum is presented and represents the assumed redshift used to shift the spectrum.

3.3.3 Derived.varAmpl:real

This field is a dimensionless value indicating the variability amplitude as a fraction of the mean. The value must be positive, but is otherwise unbound. It is a characteristic amplitude, a precise value is not required. (e.g. a value of 0.2 implies a 20 percent variation around the mean).

4 Observation-Experiment

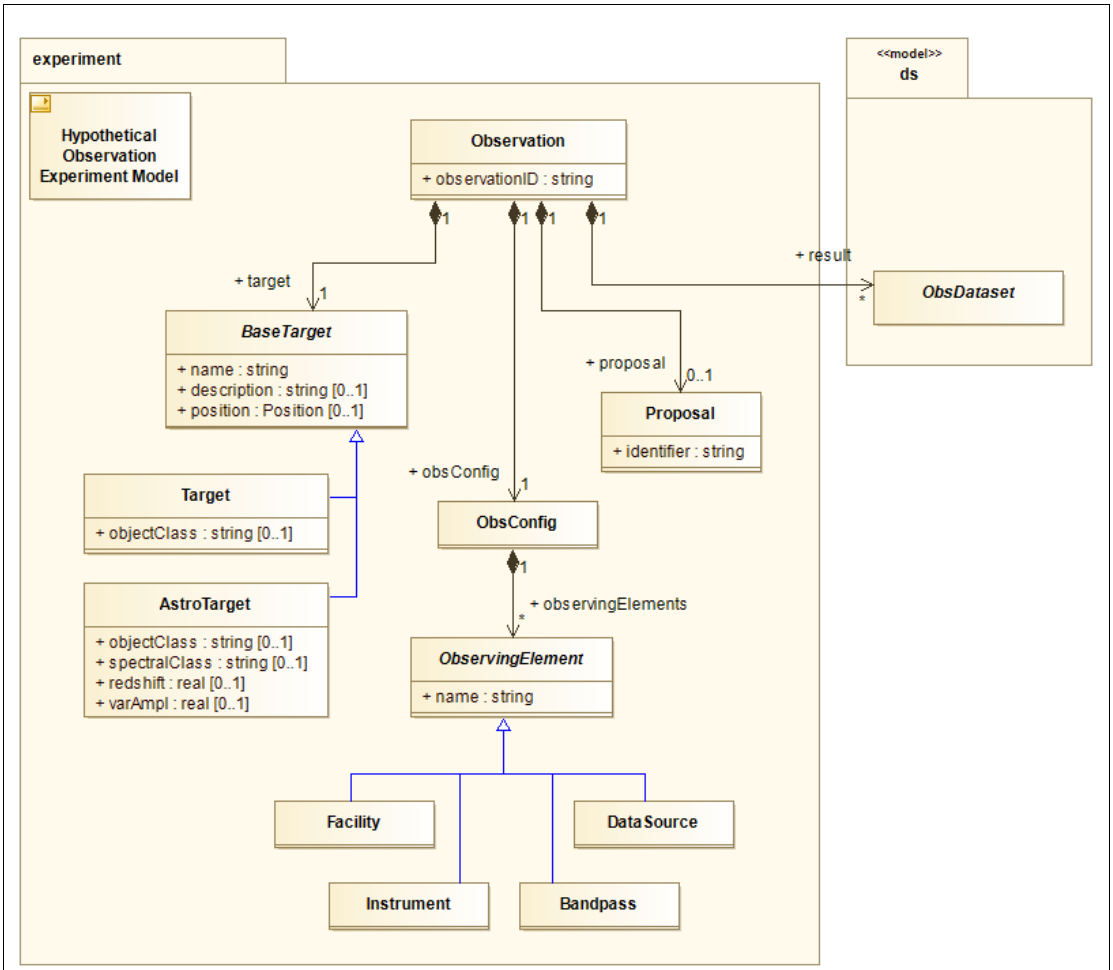


Figure 9: Observation Experiment detail diagram

The ObsDataset model refers to several elements related to an Observation and its configuration. As of the time of this writing, there is no IVOA recommendation for a general Observation data model. In lieu of this standard, this document defines a straw-man Observation model.

The Observation is modeled as a type of 'Experiment', with some basic structure defined to provide metadata about the observation target and configuration. The product, or 'result' of the Observation is zero (0) or more ObsDataset objects. This pattern is inspired by, and compatible with the Simulation Data Model [6], where a 'Simulation' can be considered another form of 'Experiment' or perhaps even another form of 'Observation'.

4.1 Observation

4.1.1 Observation.observationID:string

Internal ID determined by the data provider to uniquely identify the observation within the institution or entity performing the observation.

4.1.2 Observation.target:BaseTarget

The target of the observation. The content of this object may vary greatly depending on the goals and nature of the observation. For example the 'target' could be a galaxy, stellar object, planet, or calibration source. As such, we allow the BaseTarget class here, and permit users to define and use more content rich flavors according to their needs. The BaseTarget object content is defined in Section 4.2.

4.1.3 Observation.obsConfig:ObsConfig

Observation configuration metadata, provides information about who, where, and how the observation was conducted. The content of the ObsConfig object is given in Section 4.5.

4.1.4 Observation.proposal:Proposal

Identifies any proposal related to the observation. This field may be used to gather all observations and products related to a particular proposal. The content of the Proposal object is given in Section 4.7.

4.2 BaseTarget

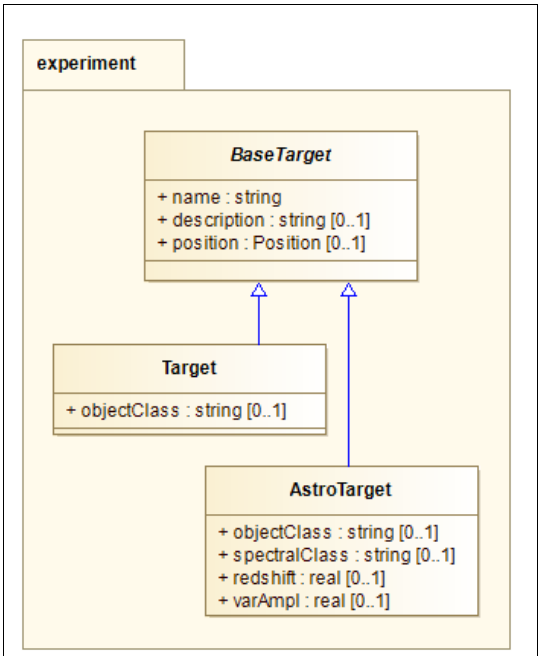


Figure 10: Target hierarchy detail diagram

Abstract base class for the Target object tree. The target object provides identifying metadata related to the subject or goal of the experiment. For an Observational experiment, this would typically be an astronomical object. The BaseTarget class defines high-level identifying information, and must be extended for particular classes of Target which may define additional characteristics.

4.2.1 BaseTarget.name:string

The target name. The primary purpose of this field is to provide the user with a recognizable identity of the particular subject or goal. However, since this may be a query-able field in data discovery protocols, care should be taken to use values which follow conventions for the domain appropriate for the data. For an astronomical object, this may be a name suitable for use within a name resolver. Simulated data might also use this sort of name (if simulating a particular object), or a more generic term such as "G2V star".

4.2.2 BaseTarget.description:string

Free form description of target.

4.2.3 BaseTarget.position:Position

This field provides the spatial location of the target. The value is a STC Position object which supports all required dimensionality and coordinate frame specification needs.

4.3 Target

Extension of BaseTarget, this is a general purpose Target object.

4.3.1 Target.objectClass:string

General classification or type of the target. This field supports the discovery of data pertaining to a common class. e.g. "Star", "Galaxy", "AGN". At the time of this writing, there is no IVOA recommended vocabulary for this field. The SIMBAD and NED databases use defined vocabularies for astronomical object classifications which may serve as the basis for such.

4.4 AstroTarget

Extension of BaseTarget specialized for astronomical objects. The AstroTarget defines additional astronomical properties of the target.

4.4.1 AstroTarget.name:string

When referring to an astronomical target, one may specify a particular object, or a more general target such as the name of a survey field. When specifying a particular object, it is highly recommended to use a name suitable for input to a name resolver.

4.4.2 AstroTarget.position:Position

In the context of the astronomical target, this field gives the nominal RA and Dec location for the target. For example, the catalog position of the source.

4.4.3 AstroTarget.objectClass:string

General classification or type of the target. This field supports the discovery of data pertaining to a common class. e.g. "Star", "Galaxy", "AGN". At the time of this writing, there is no IVOA recommended vocabulary for this field. The SIMBAD and NED databases use defined vocabularies for astronomical object classifications which may serve as the basis for such.

4.4.4 AstroTarget.spectralClass:string

Spectral class of the object. As with objectClass, there is no IVOA recommended vocabulary for specifying the spectral class of an object. There is an IVOA Note on the subject entitled "An encoding system to represent stellar spectral classes in archival databases and catalogs"[7], describing an encoding system which has been adopted by the MAST archive.

4.4.5 AstroTarget.redshift:real

This field gives the actual redshift of the astronomical object. It is normally used to store the cosmological redshift of extragalactic objects, although it may also be used to store the observed redshift of Galactic sources if that information is felt by the data provider to be useful.

Note: This is distinct from the Derived.Redshift which indicates a measured redshift value.

4.4.6 AstroTarget.varAmpl:real

Typical target variability amplitude.

4.5 ObsConfig

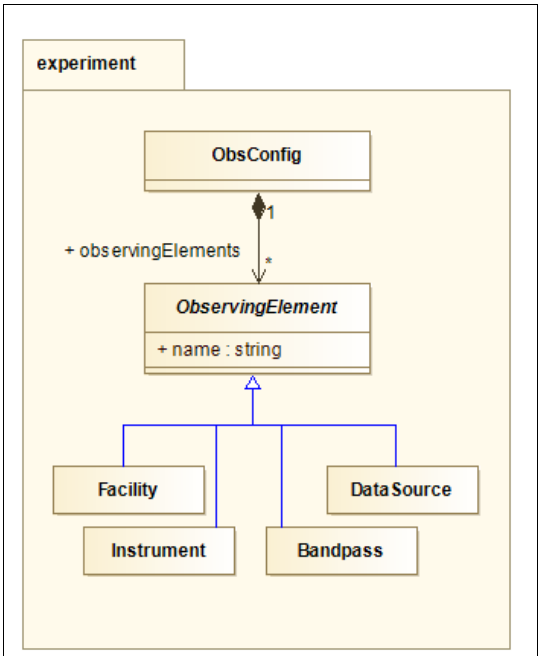


Figure 11: ObsConfig detail diagram

ObsConfig is a simple container object for all Observation Configuration metadata. It is modeled as a simple list of observing elements. Each ObservingElement provides metadata describing a particular domain of the observation setup. For example, the Facility performing the observation, the Instrument being used, etc.

4.5.1 ObsConfig.observingElements

Collection of zero (0) or more ObservingElement objects which define observation parameters.

4.6 ObservingElement

Abstract base class for defining observation parameters. Each subclass of ObservationElement pertains to a particular 'domain' of the observation setup, e.g. the Facility, or the Instrument, and may have additional structure to provide all relevant metadata in that domain.

4.6.1 ObservingElement.name:string

The name attribute identifies the particular instance of the ObservingElement. e.g. the Facility name.

4.6.2 ObservingElement subclasses

There are currently 3 empty extensions of ObservingElement, each identifying a particular domain of Observation Configuration metadata (Facility, Instrument, Bandpass, DataSource).

4.6.2.1 Facility.name:

Name of the facility performing the observation.

4.6.2.2 Instrument.name:

This field identifies the instrument used to create the data. (RM:Collection.Instrument) This can be a specific instrument name, general type or something else, e.g. a program in the case of theoretical data. We restrict this field to a single value.

4.6.2.3 Bandpass.name:

A string describing the spectral range. This field corresponds to both the Coverage.Spectral and Coverage.Spectral.Bandpass fields of the Resource Metadata document. The value may be any one of the strings listed for Coverage.Spectral (e.g. "Optical") or Coverage.Spectral.BandPass (e.g. "B"). Note that at this time, there is no fixed list of values for this latter group.

4.6.2.4 DataSource.name:

Describes the original source of the data (e.g. "survey")

4.7 Proposal

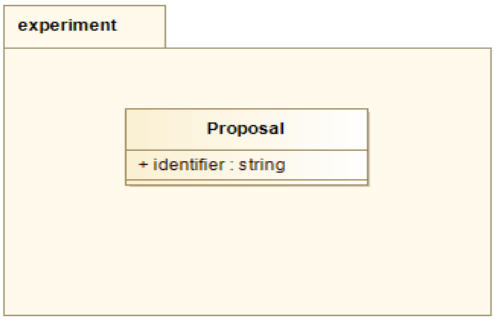


Figure 12: Proposal detail diagram

Metadata related to the proposal or document which spawned the observation.

4.7.1 Proposal.identifier:string

Tag used to uniquely identify a particular proposal within the institution or entity.

5 Data Types

5.1 IVOA Data Types

The ivoa model provides a set of standardized primitive data types as well as types for representing quantities (values with associated units and ucd). We provide a diagram of the model here, and refer the reader to Appendix F of the VO-DML modeling specification document[8] for more information.

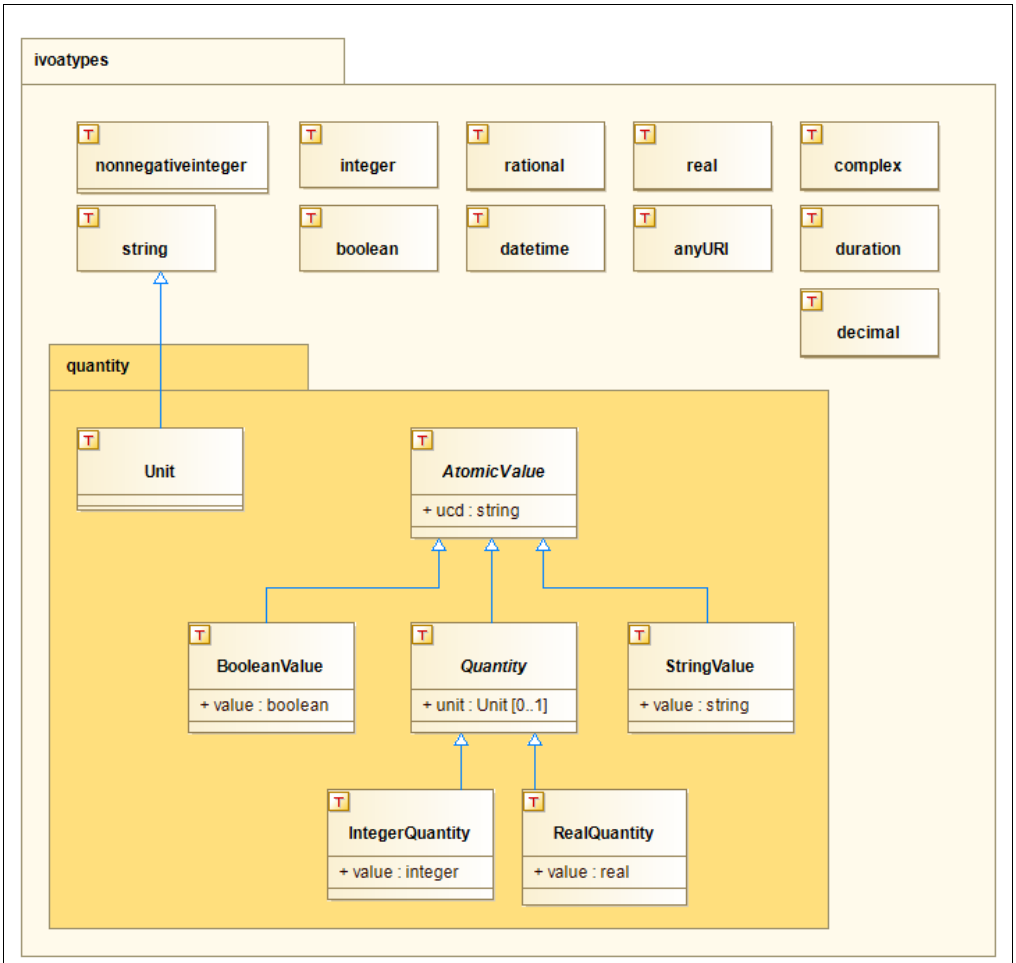


Figure 13: IVOA Base Data Types

5.1.1 Units

This model requires the use of the IVOA VOUnits Standard[9] for representing units of physical quantities. This standard reconciles common practices and current standards for use within the IVOA community.

5.1.2 UCDs

This model requires the ucd field to comply with syntax defined in ”An IVOA Standard for Unified Content Descriptors”[11].

5.1.3 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.

5.2 Dataset Model DataTypes

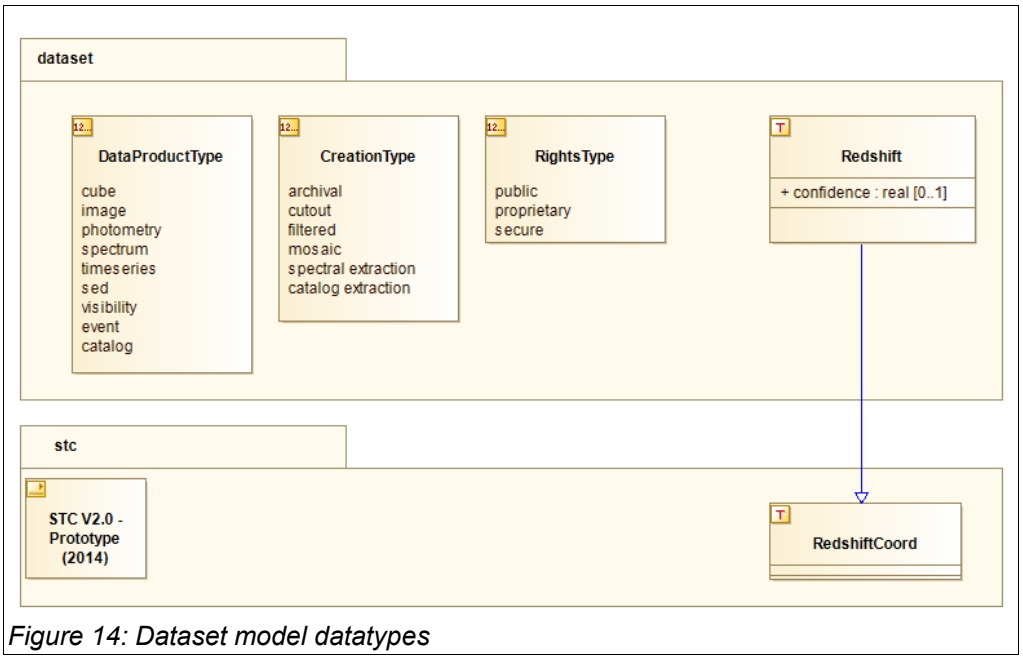


Figure 14: Dataset model datatypes

5.2.1 DataProductType

Enumeration identifying the high level classification of a data product. Allowed values are:

Token	Meaning
cube	A multidimensional astronomical image of three (3) or more axes.
image	A two (2) dimensional astronomical image.
photometry	Dataset with spectral coverage with irregular gaps.
spectrum	Dataset where spectral coverage is the primary attribute, in contiguous bins. e.g. a 1D spectrum or a long slit spectrum.
timeseries	Dataset presenting some quantity varying as a function of time. A light curve is a typical

	example of a timeseries dataset.
sed	A spectral energy distribution, an advanced data product often produced by combining data from multiple observations.
visibility	A visibility (radio) dataset. Typically this is instrumental data, and is often a complex object containing multiple files or other substructures. A visibility dataset may contain data with spatial, spectral, time, and polarization information for each measured visibility.
event	An event counting dataset (e.g. X-ray). An event dataset may contain data with spatial, spectral, and time information for each measured event.
catalog	A catalog.

5.2.2 CreationType

Enumeration of dataset creation types. Allowed values are:

Token	Meaning
archival	Indicates that it is one of a collection of datasets generated in a systematic, homogeneous way and is stored statically (or at least versioned). It will be possible to regenerate this dataset at a later date. The remaining types imply on-the-fly manipulation.
cutout	Indicates that the dataset was created "on-the-fly", by subsetting, but not by modifying values.
filtered	May involve excluding data prior to binning into samples, also "on-the-fly"
mosaic	Combines multiple original datasets "on-the-fly"
spectral extraction	Has been extracted, for example, from a spectral data cube.
catalog extraction	Has been extracted from a catalog.

5.2.3 RightsType

Enumeration indicating access rights levels. Allowed values are:

Token	Meaning
public	unrestricted, public access is allowed, without authentication.
secure	authenticated, public access is allowed.
proprietary	only proprietary access is allowed with authentication.

5.2.4 Redshift

Extension of the STC RedshiftCoord type. Provides a measured value along the Redshift coordinate axis.

5.2.4.1 Redshift.coords:Quantity

Defined as $\Delta\lambda/\lambda$ and may be positive or negative.

5.2.4.2 Redshift.confidence:real

Confidence is an additional measure of accuracy. A probability between 0 and 1 that the quoted errors do apply. Example: This measure is used in the Sloan spectral service to provide a way of describing the estimated probability that the redshift is completely in error because the lines have been misidentified. Its default value is 1.0.

6 STC 2.0 Prototype Data Model

At the time of this writing, the IVOA data model work is formalizing a revision of the current STC recommendation (1.33). This new STC model is targeted for use within this document and for the N-Dimensional Cube model. We provide here, a prototype definition of the STC-2.0 data model which is a best approximation of the content and structure of this work. The objects and types presented here are sufficient to support the development of this work, but are not comprehensive of the scope and content of the STC model.

This section should be removed upon the completion of the STC-2.0 model, and related documents updated to that specification.

6.1 Coordinate Frames

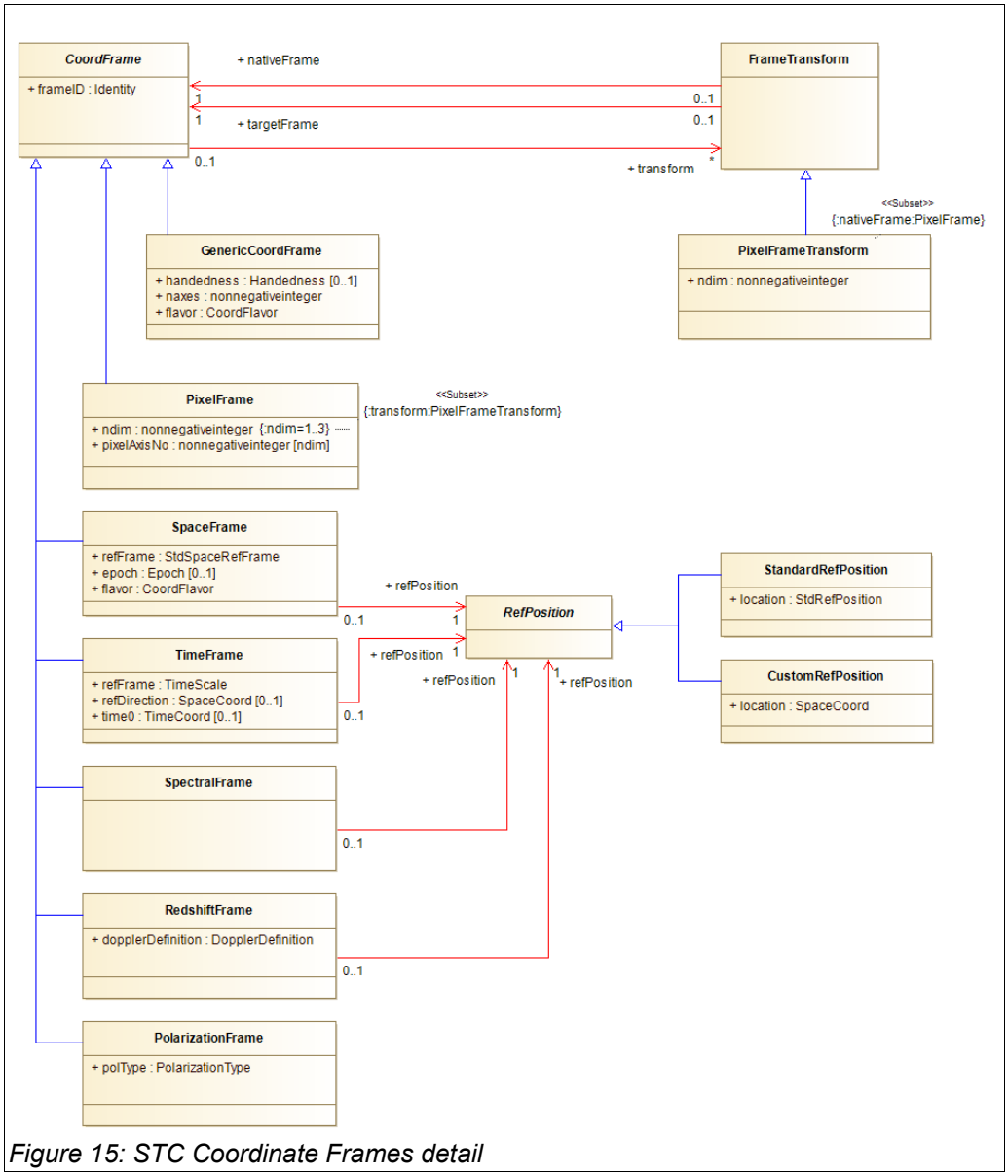


Figure 15: STC Coordinate Frames detail

6.1.1 CoordFrame

Abstract head of the Coordinate Frame hierarchy. A Coordinate frames define the domain in which other objects reside. We provide specialized frames for each of the primary astronomical domains, and a generic frame which may be used for domains not represented here.

6.1.1.1 CoordFrame.frameID:Identity

All coordinate frames have an identifier which is unique within the context of their usage (for example a Dataset).

6.1.1.2 CoordFrame.transform:FrameTransform

Reference to one (1) or more FrameTransform objects, in which the frame is a member. This reference enables objects which specify their frames to have direct access to Transform information enabling them to be converted to other frames.

NOTE: This attribute is under discussion.

6.1.2 GenericCoordFrame

Generic coordinate frame for use describing domains not represented by the specialized frames.

6.1.2.1 GenericCoordFrame.handedness:Handedness

Handedness ("left" or "right") of the frame.

6.1.2.2 GenericCoordFrame.naxes:nonnegativeinteger

Number of axes in the domain.

6.1.2.3 GenericCoordFrame.flavor:CoordFlavor

Specifies the structure of the frame (e.g. Cartesian, Spherical, Polar, etc.).

6.1.3 PixelFrame

Specifies a set of pixel axes which are considered a unit. For example, 2 pixel axes which map to a 2-D SpaceFrame.

6.1.3.1 PixelFrame.ndim:nonnegativeinteger

Dimensionality of the pixel frame. A pixel frame is constrained to be either 1-, 2-, or 3-D.

6.1.3.2 PixelFrame.pixelAxisNo:nonnegativeinteger

This specifies which pixel axis each dimension of the frame corresponds to.

6.1.4 SpaceFrame

Coordinate frame for the Spatial domain.

6.1.4.1 SpaceFrame.refFrame:StdSpaceRefFrame

Spatial reference frame specifying the orientation of the frame.

6.1.4.2 SpaceFrame.refPosition:RefPosition

Origin of the spatial coordinate.

6.1.4.3 SpaceFrame.epoch:Epoch

6.1.4.4 SpaceFrame.flavor:CoordFlavor

6.1.5 TimeFrame

Coordinate frame for the Time domain.

6.1.5.1 TimeFrame.refFrame:TimeScale

IAU recognized time scale.

6.1.5.2 TimeFrame.refPosition:RefPosition

Origin of the time coordinate frame.

6.1.5.3 TimeFrame.refDirection:SpaceCoord

Direction of origin.

6.1.5.4 TimeFrame.time0:TimeCoord

6.1.6 SpectralFrame

Coordinate frame for the Spectral domain.

6.1.6.1 SpectralFrame.refPosition:RefPosition

Origin of the Spectral coordinate frame.

6.1.7 RedshiftFrame

Coordinate frame for the Redshift domain.

6.1.7.1 RedshiftFrame.refPosition:RefPosition

Origin of the Redshift coordinate frame.

6.1.7.2 RedshiftFrame.dopplerDefinition:DopplerDefinition

Specifies what the definition of redshift is, and how it should be translated to doppler velocity.

6.1.8 PolarizationFrame

Coordinate frame for the Polarization domain.

6.1.8.1 PolarizationFrame.polType:PolarizationType

Specifies the type of polarization.

6.1.9 RefPosition

Abstract head of the reference position objects.

6.1.10 StandardRefPosition

Standard reference position, represented by an enumeration.

6.1.10.1 StandardRefPosition.location:StdRefPosition

6.1.11 CustomRefPosition

Custom reference position, represented by a spatial coordinate.

6.1.11.1 CustomRefPosition.location:SpaceCoord

6.2 Coordinate Systems

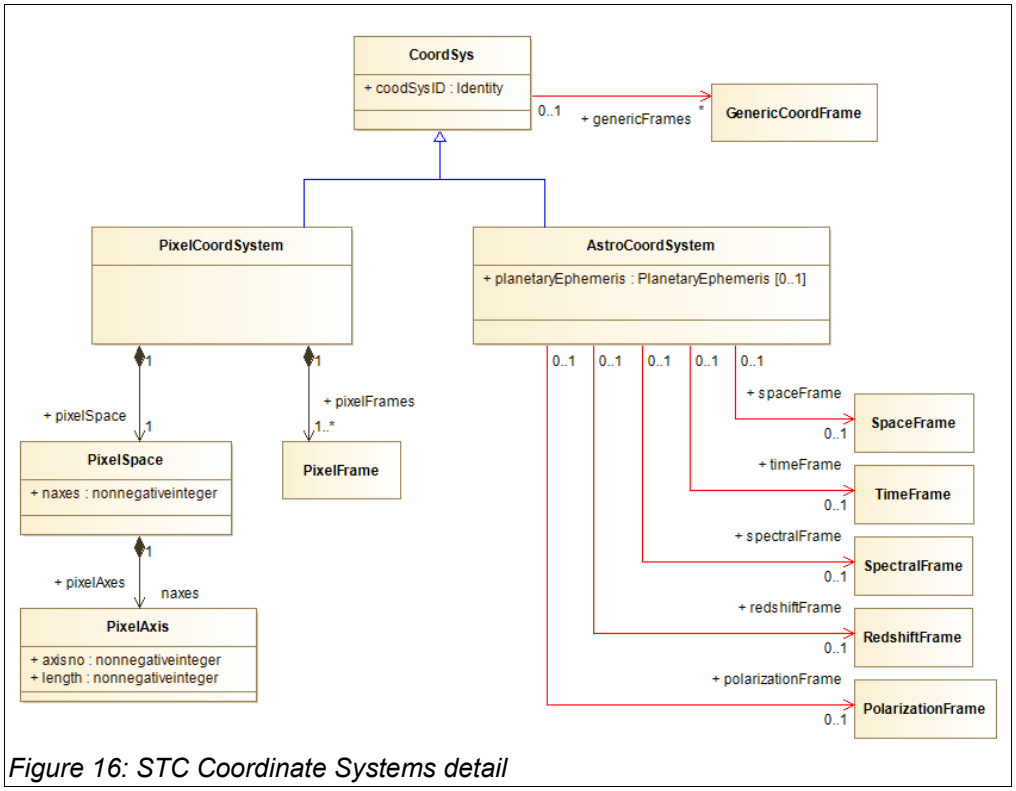


Figure 16: STC Coordinate Systems detail

6.2.1 CoordSys

Coordinate Systems are a collection of coordinate frames which completely define the space in all domains.

6.2.1.1 CoordSys.coordSysID:Identity

6.2.1.2 CoordSys.genericFrames:GenericCoordFrame

Zero (0) or more coordinate frames not represented by the specialized frames.

6.2.2 PixelCoordSystem

6.2.2.1 PixelCoordSystem.pixelSpace:PixelSpace

Describes the number and span of each pixel axis.

6.2.2.2 PixelCoordSystem.pixelFrames:PixelFrame

Collection of one (1) or more pixel frames defining the pixel coordinate system.

6.2.3 AstroCoordSystem

Extension of CoordSys specialized for astronomical systems. This container holds a reference to zero (0) or one (1) of each specialized domain frame.

6.2.3.1 AstroCoordSystem.planetaryEphemeris:PlanetaryEphemeris

6.2.3.2 AstroCoordSystem.spaceFrame:SpaceFrame

Reference to zero (0) or one (1) Space frame.

6.2.3.3 AstroCoordSystem.timeFrame:TimeFrame

Reference to zero (0) or one (1) Time frame.

6.2.3.4 AstroCoordSystem.spectralFrame:SpectralFrame

Reference to zero (0) or one (1) Spectral frame.

6.2.3.5 AstroCoordSystem.redshiftFrame:RedshiftFrame

Reference to zero (0) or one (1) Redshift frame.

6.2.3.6 AstroCoordSystem.polarizationFrame:PolarizationFrame

Reference to zero (0) or one (1) Polarization frame.

6.2.4 PixelSpace

This object defines the number and size of each pixel axis

6.2.4.1 PixelSpace.naxes:nonnegativeinteger

Number of pixel axes.

6.2.4.2 PixelSpace.pixelAxes:PixelAxis

Axis number and size. One per axis.

6.2.5 PixelAxis

This object identifies a pixel axis and defines its size.

6.2.5.1 PixelAxis.axisno:nonnegativeinteger

Axis number

6.2.5.2 PixelAxis.length:nonnegativeinteger

Length of the pixel axis.

6.3 Coordinates

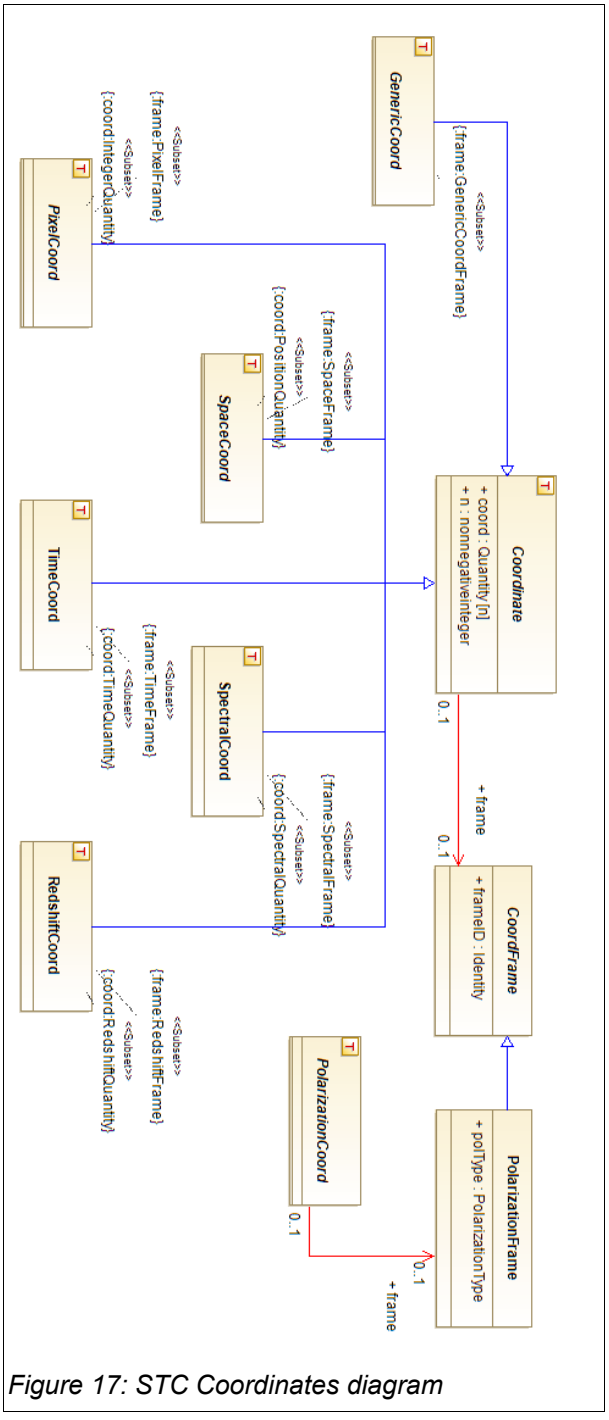


Figure 17: STC Coordinates diagram

6.3.1 Coordinate

Abstract head of the Coordinate hierarchy. This class defines the base coordinate type in very generic terms. This allows all coordinates to share a common interface at the most basic level. Extensions of this class may specify appropriate constraints to the dimensionality (n), value type, and frame type to suit the requirements of the domain.

6.3.1.1 Coordinate.n:nonnegativeinteger

Number of coordinate values, this attribute represents the dimensionality of the coordinate.

6.3.1.2 Coordinate.coord:RealQuantity

Array of 'n' coordinate values.

6.3.1.3 Coordinate.frame:CoordFrame

Reference to zero (0) or one (1) CoordFrame describing the domain of the coordinate.

6.3.2 GenericCoord

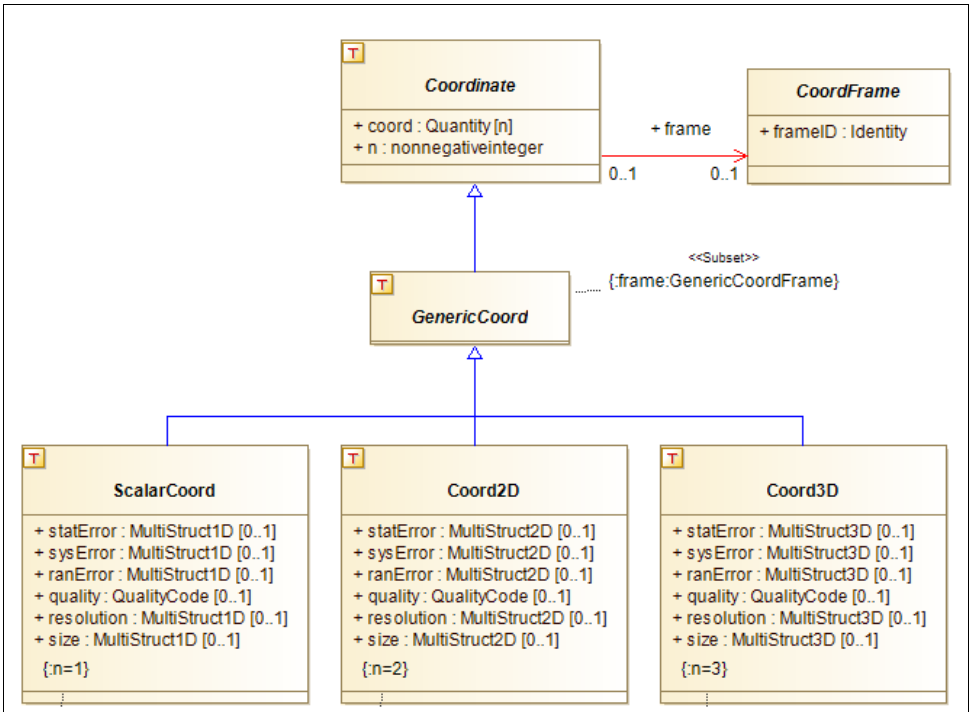


Figure 18: Generic Coordinates diagram

Coordinate type for use when a specialized domain type does not apply.

6.3.2.1 GenericCoord.frame:GenericCoordFrame

The GenericCoord subsets, or constrains, the frame attribute to the generic frame type. Reference to zero (0) or one (1) GenericCoordFrame describing the domain of the coordinate.

6.3.3 ScalarCoord

Extension of GenericCoord for a 1D coordinate, adding 1D error attributes as well as resolution and size.

6.3.3.1 ScalarCoord.n:nonnegativeinteger

Coordinate dimensionality, constrained to 1.

6.3.3.2 ScalarCoord.statError:MultiStruct1D

Statistical error associated with the coordinate value, if any, is provided by zero (0) or one (1) MultiStruct objects which facilitate a variety of representations of the error with respect to the value.

6.3.3.3 ScalarCoord.sysError:MultiStruct1D

Systematic error associated with the coordinate value, if any, is provided by zero (0) or one (1) MultiStruct objects which facilitate a variety of representations of the error with respect to the value.

6.3.3.4 ScalarCoord.ranError:MultiStruct1D

Other random error associated with the coordinate value, if any, is provided by zero (0) or one (1) MultiStruct objects which facilitate a variety of representations of the error with respect to the value.

6.3.3.5 ScalarCoord.quality:QualityCode

General statement of quality or reliability of the coordinate.

6.3.3.6 ScalarCoord.resolution:MultiStruct1D

Coordinate resolution. (TBR)

6.3.3.7 ScalarCoord.size:MultiStruct1D

Size (TBR)

6.3.4 Coord2D

Extension of GenericCoord for a 2D coordinate, adding 2D error attributes as well as resolution and size.

6.3.4.1 Coord2D.n:nonnegativeinteger

Coordinate dimensionality, constrained to 2.

6.3.4.2 Coord2D.statError:MultiStruct2D

Statistical error associated with the coordinate value, if any, is provided by zero (0) or one (1) MultiStruct objects which facilitate a variety of representations of the error with respect to the value.

6.3.4.3 Coord2D.sysError:MultiStruct2D

Systematic error associated with the coordinate value, if any, is provided by zero (0) or one (1) MultiStruct objects which facilitate a variety of representations of the error with respect to the value.

6.3.4.4 Coord2D.ranError:MultiStruct2D

Other random error associated with the coordinate value, if any, is provided by zero (0) or one (1) MultiStruct objects which facilitate a variety of representations of the error with respect to the value.

6.3.4.5 Coord2D.quality:QualityCode

General statement of quality or reliability of the coordinate.

6.3.4.6 Coord2D.resolution:MultiStruct2D

Coordinate resolution. (TBR)

6.3.4.7 Coord2D.size:MultiStruct2D

Size (TBR)

6.3.5 Coord3D

Extension of GenericCoord for a 3D coordinate, adding 3D error attributes as well as resolution and size.

6.3.5.1 Coord3D.n:nonnegativeinteger

Coordinate dimensionality, constrained to 3.

6.3.5.2 Coord3D.statError:MultiStruct3D

Statistical error associated with the coordinate value, if any, is provided by zero (0) or one (1) MultiStruct objects which facilitate a variety of representations of the error with respect to the value.

6.3.5.3 Coord3D.sysError:MultiStruct3D

Systematic error associated with the coordinate value, if any, is provided by zero (0) or one (1) MultiStruct objects which facilitate a variety of representations of the error with respect to the value.

6.3.5.4 Coord3D.ranError:MultiStruct3D

Other random error associated with the coordinate value, if any, is provided by zero (0) or one (1) MultiStruct objects which facilitate a variety of representations of the error with respect to the value.

6.3.5.5 Coord3D.quality:QualityCode

General statement of quality or reliability of the coordinate.

6.3.5.6 Coord3D.resolution:MultiStruct3D

Coordinate resolution. (TBR)

6.3.5.7 Coord3D.size:MultiStruct3D

Size (TBR)

6.3.6 SpaceCoord

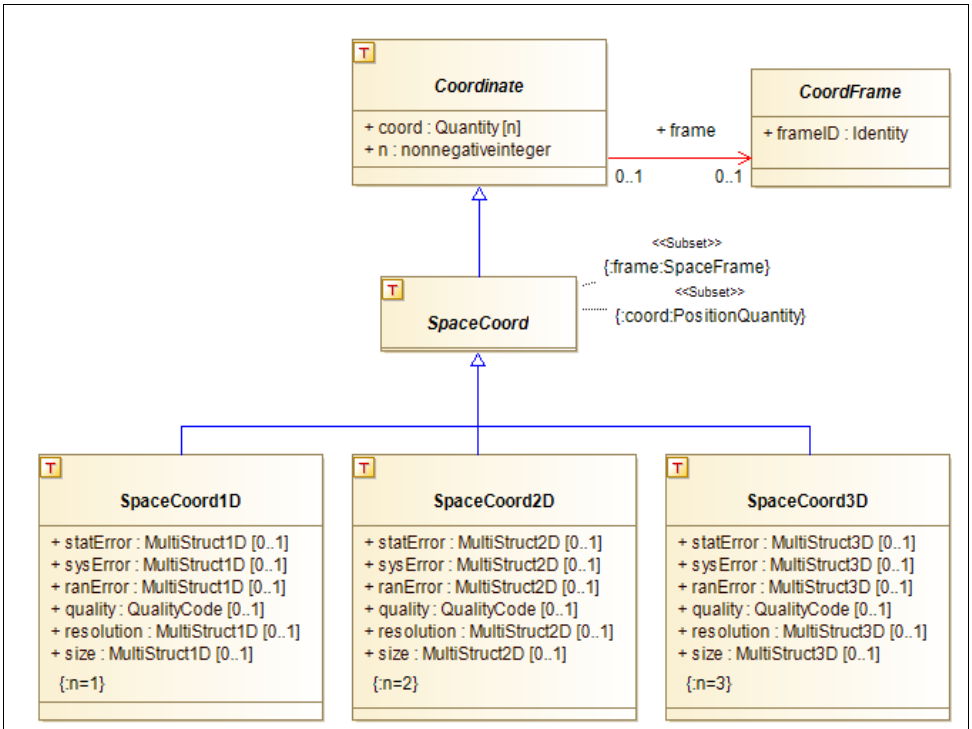


Figure 19: Space Domain Coordinates diagram

Abstract extension of Coordinate type specialized to the Space domain. This class is responsible for ensuring that all quantities and errors are compatible with this domain.

6.3.6.1 SpaceCoord.frame:SpaceFrame

The SpaceCoord subsets, or constrains, the frame attribute to the type appropriate for the Space domain. Reference to zero (0) or one (1) SpaceFrame describing the domain of the coordinate.

6.3.6.2 SpaceCoord.coord:PositionQuantity

The SpaceCoord subsets, or constrains, the coordinate value to a PositionQuantity which enforces that the associated units and ucds are compatible with the spatial domain.

6.3.7 SpaceCoord1D

Extension of SpaceCoord for 1-D Spatial coordinates. This object adds the same attribute profile as the ScalarCoord described in section 6.3.3.

6.3.8 SpaceCoord2D

Extension of SpaceCoord for 2-D Spatial coordinates. This object adds the same attribute profile as the Coord2D described in section 6.3.4.

6.3.9 SpaceCoord3D

Extension of SpaceCoord for 3-D Spatial coordinates. This object adds the same attribute profile as the Coord3D described in section 6.3.5.

6.3.10 TimeCoord

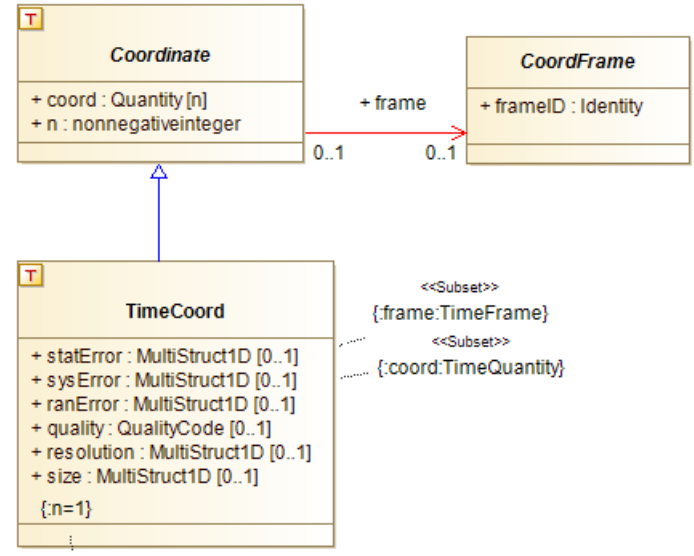


Figure 20: Time Domain Coordinates diagram

Extension of Coordinate type specialized to the Time domain. This class is responsible for ensuring that all quantities and errors are compatible with this domain. This object adds the same attribute profile as the ScalarCoord described in section 6.3.3.

6.3.10.1TimeCoord.frame:TimeFrame

The TimeCoord subsets, or constrains, the frame attribute to the type appropriate for the Time domain. Reference to zero (0) or one (1) TimeFrame describing the domain of the coordinate.

6.3.10.2TimeCoord.coord:TimeQuantity

The TimeCoord subsets, or constrains, the coordinate value to a TimeQuantity which enforces that the associated units and ucds are compatible with the time domain.

6.3.11 SpectralCoord

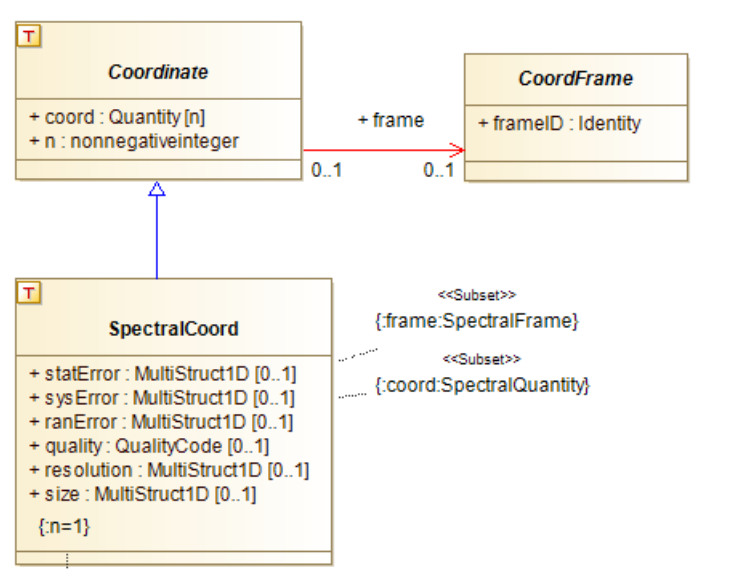


Figure 21: Spectral Domain Coordinates diagram

Extension of Coordinate type specialized to the Spectral domain. This class is responsible for ensuring that all quantities and errors are compatible with this domain. This object adds the same attribute profile as the ScalarCoord described in section 6.3.3.

6.3.11.1 SpectralCoord.frame:SpectralFrame

The SpectralCoord subsets, or constrains, the frame attribute to the type appropriate for the Spectral domain. Reference to zero (0) or one (1) SpectralFrame describing the domain of the coordinate.

6.3.11.2 SpectralCoord.coord:SpectralQuantity

The SpectralCoord subsets, or constrains, the coordinate value to a SpectralQuantity which enforces that the associated units and ucds are compatible with the spectral domain.

6.3.12 RedshiftCoord

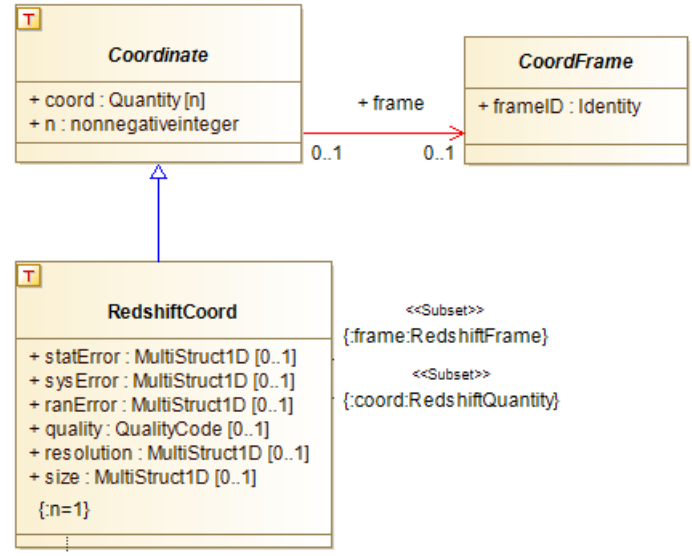


Figure 22: Time Domain Coordinates diagram

Extension of Coordinate type specialized to the Redshift domain. This class is responsible for ensuring that all quantities and errors are compatible with this domain. This object adds the same attribute profile as the ScalarCoord described in section 6.3.3.

6.3.12.1RedshiftCoord.frame:RedshiftFrame

The RedshiftCoord subsets, or constrains, the frame attribute to the type appropriate for the Redshift domain. Reference to zero (0) or one (1) RedshiftFrame describing the domain of the coordinate.

6.3.12.2RedshiftCoord.coord:RedshiftQuantity

The RedshiftCoord subsets, or constrains, the coordinate value to a RedshiftQuantity which enforces that the associated units and ucds are compatible with the redshift domain.

6.3.13 PixelCoord

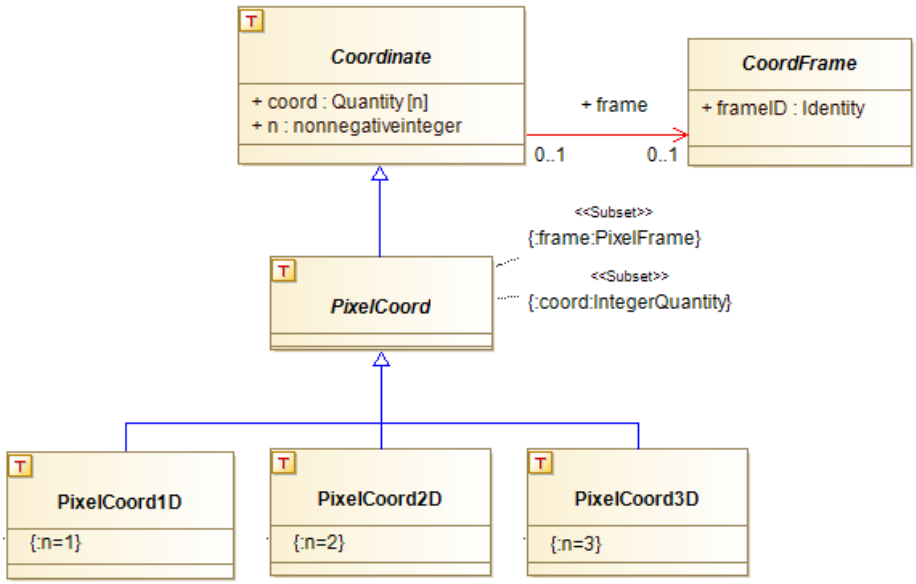


Figure 23: Pixel Coordinates diagram

PixelCoord is an extension of Coordinate for use in the Pixel domain. The coordinate value of the pixel classes is restricted to an IntegerQuantity, and the frame reference to a PixelFrame type. While, in general, a pixel coordinate may be of any dimension, we define 1-D, 2-D, and 3-D flavors of PixelCoord which corresponds to the dimensionality of the PixelFrames defined in this model.

6.3.14 PixelCoord1D

Extension of PixelCoord for 1-D Pixel coordinates.

6.3.15 PixelCoord2D

Extension of PixelCoord for 2-D Pixel coordinates.

6.3.16 PixelCoord3D

Extension of PixelCoord for 3-D Pixel coordinates.

6.3.17 PolarizationCoord

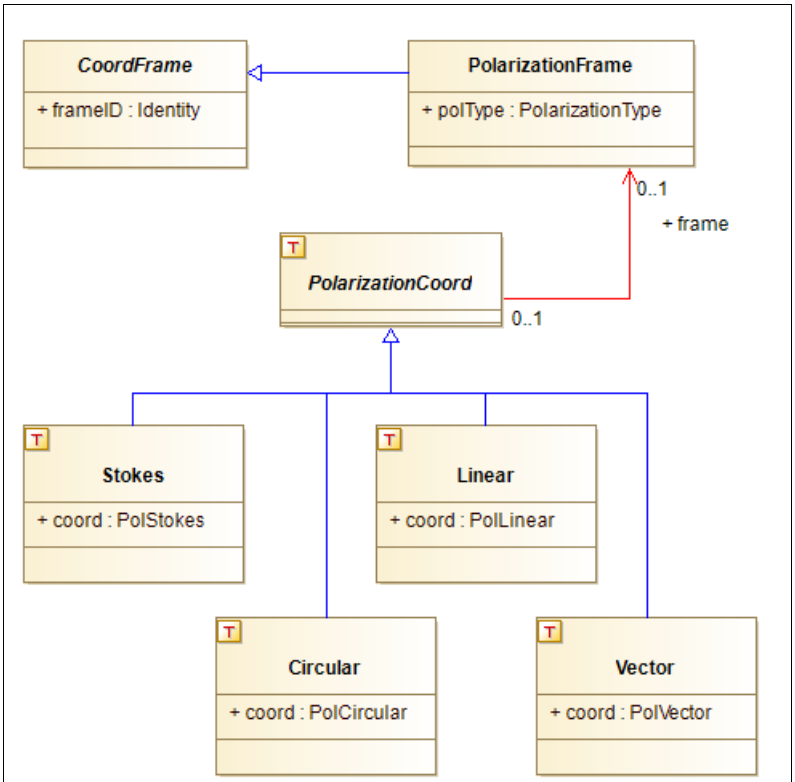


Figure 24: Generic Coordinates diagram

Abstract head of the Polarization coordinate type. This type is NOT an extension of the Coordinate hierarchy due to the fact that the value set for this domain are enumerated states rather than Quantities within a domain. The structure is compatible with the base Coordinate type to reflect the similar nature and usage of these types with the other domain coordinates.

6.3.17.1 PolarizationCoord.frame:PolarizationFrame

Reference to zero (0) or one (1) PolarizatonFrame describing the domain of the coordinate.

6.3.18 Stokes

Extension of PolarizationCoord specialized for the Stokes Polarization type.

6.3.18.1 Stokes.coord:PolStokes

Coordinate value is one of the enumerated set defined by PolStokes

6.3.19 Circular

Extension of PolarizationCoord specialized for the Circular Polarization type.

6.3.19.1 Circular.coord:PolCircular

Coordinate value is one of the enumerated set defined by PolCircular

6.3.20 Linear

Extension of PolarizationCoord specialized for the Linear Polarization type.

6.3.20.1 Linear.coord:PolLinear

Coordinate value is one of the enumerated set defined by PolLinear

6.3.21 Vector

Extension of PolarizationCoord specialized for the Vector Polarization type.

6.3.21.1 Vector.coord:PolVector

Coordinate value is one of the enumerated set defined by PolVector

6.4 MultiStruct

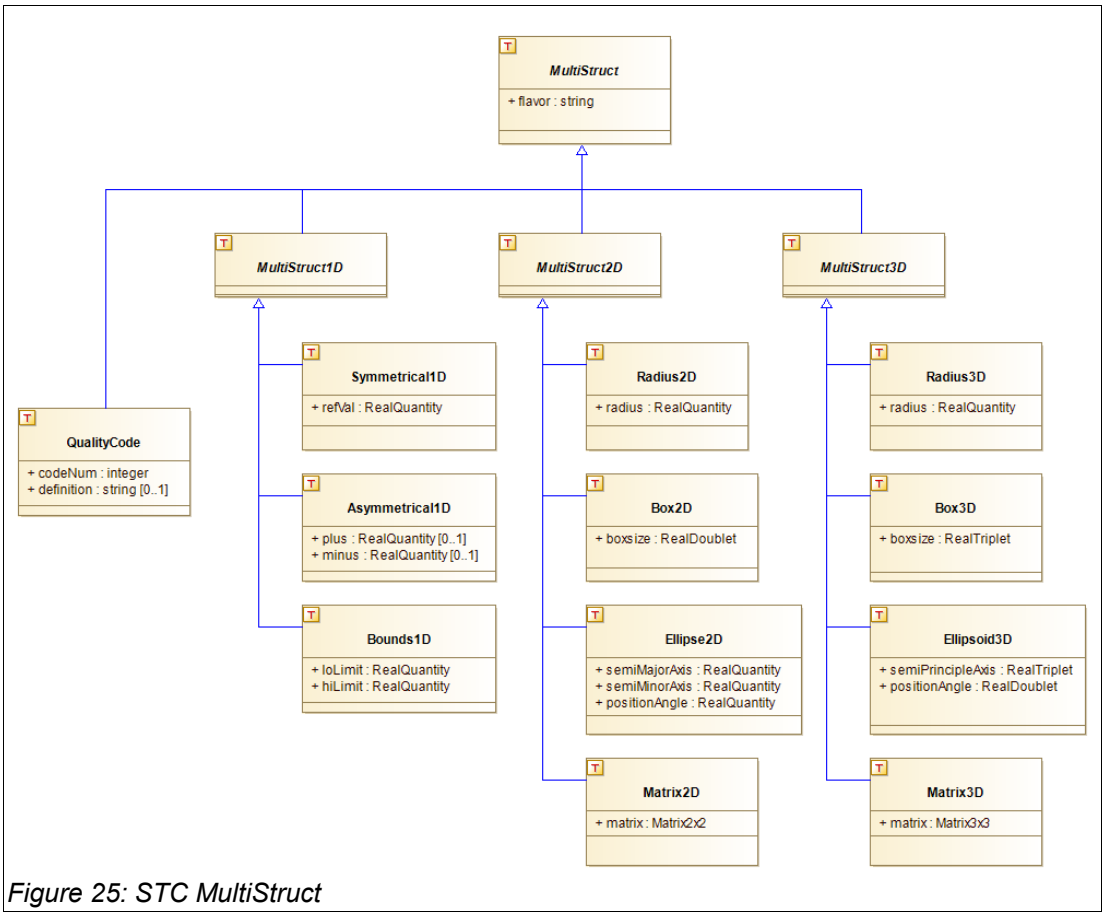


Figure 25: STC MultiStruct

6.4.1 MultiStruct

Abstract head of the MultiStruct hierarchy. This object set represents a multiple-purpose structure which is used to represent Errors, Resolutions, and Sizes in this model.

6.4.2 MultiStruct1D

Abstract head for 1-Dimensional MultiStruct objects

6.4.3 MultiStruct2D

Abstract head for 2-Dimensional MultiStruct objects

6.4.4 MultiStruct3D

Abstract head for 3-Dimensional MultiStruct objects

6.4.5 Symmetrical1D

The entity is represented by a symmetrical distribution about the associated value.

6.4.5.1 Asymmetrical.refVal:RealQuantity

6.4.6 Asymmetrical1D

The entity is represented by an asymmetrical distribution about the associated value.

6.4.6.1 Asymmetrical.plus:RealQuantity

6.4.6.2 Asymmetrical.minus:RealQuantity

6.4.7 Bounds1D

The entity is represented by a range.

6.4.7.1 Bounds.loLimit:RealQuantity

6.4.7.2 Bounds.hiLimit:RealQuantity

6.4.8 Radius2D

The entity is represented by a symmetrical distribution about the associated value.

6.4.8.1 Radius2D.radius:RealQuantity

6.4.9 Box2D

The entity is represented by a 2-D box.

6.4.9.1 Box2D.boxsize:RealDoublet

Size of box in each dimension.

6.4.10 Ellipse2D

The entity is represented by a 2-D ellipse.

6.4.10.1 Ellipse2D.semiMajorAxis:RealQuantity

6.4.10.2 Ellipse2D.semiMinorAxis:RealQuantity

6.4.10.3 Ellipse2D.positionAngle:RealQuantity

6.4.11 Matrix2D

The entity is represented by a 2-D Matrix.

6.4.11.1 Matrix2D.matrix:Matrix2x2

6.4.12 Radius3D

The entity is represented by a symmetrical distribution about the associated value.

6.4.12.1 Radius3D.radius:RealQuantity

6.4.13 Box3D

The entity is represented by a 3-D box.

6.4.13.1 Box3D.boxsize:RealTriplet

Size of box in each dimension.

6.4.14 Ellipsoid3D

The entity is represented by a 3-D ellipsoid.

6.4.14.1 Ellipsoid3D.semiPrincipleAxis:RealTriplet

6.4.14.2 Ellipsoid3D.positionAngle:RealDoublet

6.4.15 Matrix3D

The entity is represented by a 3-D Matrix.

6.4.15.1 Matrix3D.matrix:Matrix3x3

6.4.16 QualityCode

The entity is represented by a numerical code.

6.4.16.1 QualityCode.codeNum:integer

Numerical code.

6.4.16.2 QualityCode.definition:string

String definition of the code.

6.5 Transforms

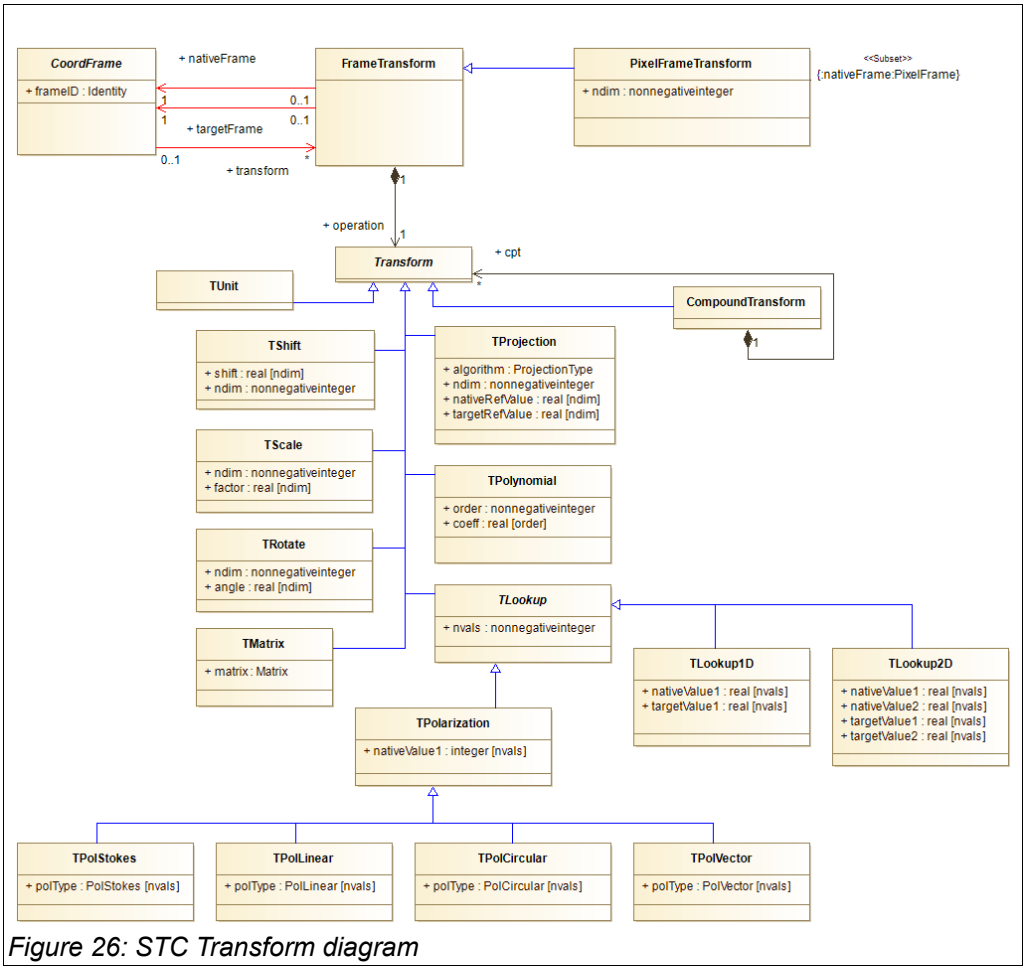


Figure 26: STC Transform diagram

6.5.1 FrameTransform

Defines the relationship between two coordinate frames. Is used in the conversion of values (Coordinates, Regions, etc) represented in one frame, to the other.

6.5.1.1 FrameTransform.nativeFrame:CoordFrame

Reference to the source or originating frame.

6.5.1.2 FrameTransform.targetFrame:CoordFrame

Reference to the target or destination frame

6.5.1.3 FrameTransform.operation:Transform

Operation(s) to peform.

6.5.2 Transform

Transforms perform mathematical operations on input object (e.g. Coordinates), but have not direct knowledge of the physical context of the object. The transform operations defined here are generalized components which can be combined to support a wide range of complex transforms. This includes, but is not limited to, the FITS Linear and WCS coordinate transforms.

6.5.3 CompoundTransform

Holds an ordered sequence of Transform operations. Enables complex transforms to be defined from the simpler components.

6.5.4 TUnit

Unit transform. Multiplication by 1.

6.5.5 TShift

Shift operation.

6.5.5.1 TShift.ndim:nonnegativeinteger

Number of dimensions.

6.5.5.2 TShift.shift:real[ndim]

Amount to shift in each dimension.

6.5.6 TScale

Multiplication operation

6.5.6.1 TScale.ndim:nonnegativeinteger

Number of dimensions.

6.5.6.2 TScale.factor:real[ndim]

Scale factor in each dimension.

6.5.7 TRotate

Angular rotation operation. NOTE: this definition is over-simplified and needs to be fleshed out with specifics about whether these are rotation angles about each axis, or a constant rotation for axis pair, etc.

6.5.7.1 TRotate.ndim:nonnegativeinteger

Number of dimensions.

6.5.7.2 TRotate.angle:real[ndim]

Angle of rotation.

6.5.8 TMatrix

Matrix operation.

6.5.8.1 TMatrix.matrix:Matrix

6.5.9 TProjection

WCS Projection operation.

6.5.9.1 TProjection.algorithm:ProjectionType

Projection algorithm to apply.

6.5.9.2 TProjection.ndim:nonnegativeinteger

Dimensionality of values.

6.5.9.3 TProjection.nativeRefValue:real[ndim]

Reference value for originating frame (e.g. Reference Pixel)

6.5.9.4 TProjection.targetRefValue:real[ndim]

Reference value at destination frame (e.g. Reference Value)

6.5.10 TPolynomial

1-D Polynomial operation

6.5.10.1TPolynomial.order:nonnegativeinteger

Polynomial order or degree

6.5.10.2TPolynomial.coeff:real[order+1]

Polynomial coefficients.

6.5.11TLookup

Abstract head of Lookup table operations.

6.5.11.1TLookup.nvals:nonnegativeinteger

Number of lookup entries provided (length of table).

6.5.12 TLookup1D

Lookup table in one dimension. Results are determined by linear interpolation of nearest neighbors in the provided lookup table data.

6.5.12.1TLookup1D.nativeValue1:real[nvals]

Originating frame values.

6.5.12.2TLookup1D.targetValue1:real[nvals]

Destination frame values.

6.5.13 TLookup2D

Lookup table in two dimensions. Results are determined by linear interpolation of nearest neighbors in the provided lookup table data.

6.5.13.1TLookup2D.nativeValue1:real[nvals]

Originating frame values for axis 1.

6.5.13.2TLookup2D.nativeValue2:real[nvals]

Originating frame values for axis 2.

6.5.13.3TLookup2D.targetValue1:real[nvals]

Destination frame values for axis 1.

6.5.13.4TLookup2D.targetValue2:real[nvals]

Destination frame values for axis 2.

6.5.14 TPolarization

Extension of TLookup, specialized for polarization data.

6.5.14.1TPolarization.nativeValue1:integer[nvals]

Numerical value to associate with a polarization state.

Two use cases are considered:

1) A numerical value is assigned for each polarization state. For example, the FITS-3.0 "Conventional Stokes values" which assign the numbers (1..4,-1..-8) to various polarization states. In this case, nvals is the number of such states and the nativeValue1 array holds the associated value.

2) A simple list of pixels, where a polarization state is assigned to each pixel. For example, an image with polarization axis, where each index corresponds to a different polarization state. In this case, nvals is the number of pixels, and the nativeValue1 array holds the pixel indexes.

6.5.15 TPolStokes

Polarization lookup where results are Stokes polarization states.

6.5.15.1TPolStokes.polType:PolStokes[nvals]

6.5.16 TPolLinear

Polarization lookup where results are Linear polarization states.

6.5.16.1TPolLinear.polType:PolLinear[nvals]

6.5.17 TPolCircular

Polarization lookup where results are Circular polarization states.

6.5.17.1TPolCircular.polType:PolCircular[nvals]

6.5.18 TPolVector

Polarization lookup where results are Vector polarization states.

6.5.18.1TPolVector.polType:PolVector[nvals]

6.6 Mappings

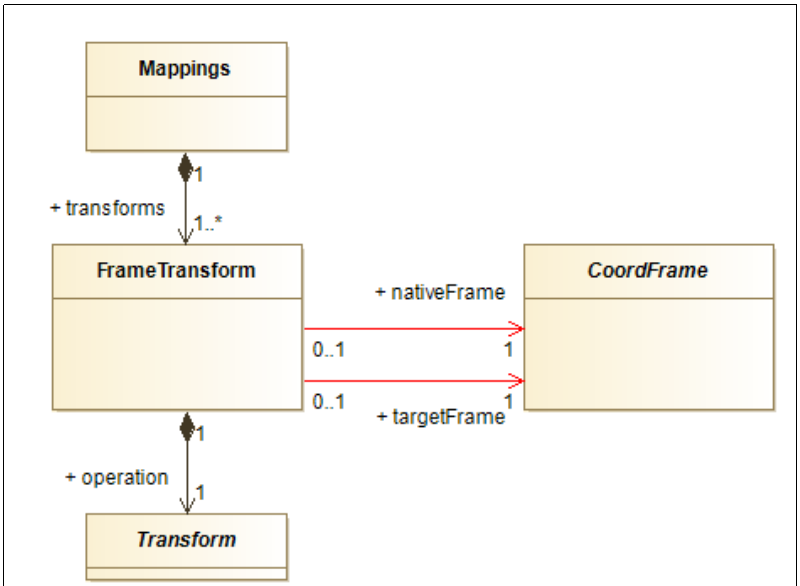


Figure 27: Mappings detail

The Mappings object is a container to composed of a set of Frame transforms. This object provides a convenient container to encapsulate a set of coordinate frame transforms associated with a particular object, such as a 2D image. It facilitates the transfer and sharing of this information between applications.

6.6.1 Mappings.transforms

The Mappings container stores one (1) or more FrameTransform objects, each of which defines a relation between two coordinate frames.

6.7 STC Model Enumerations

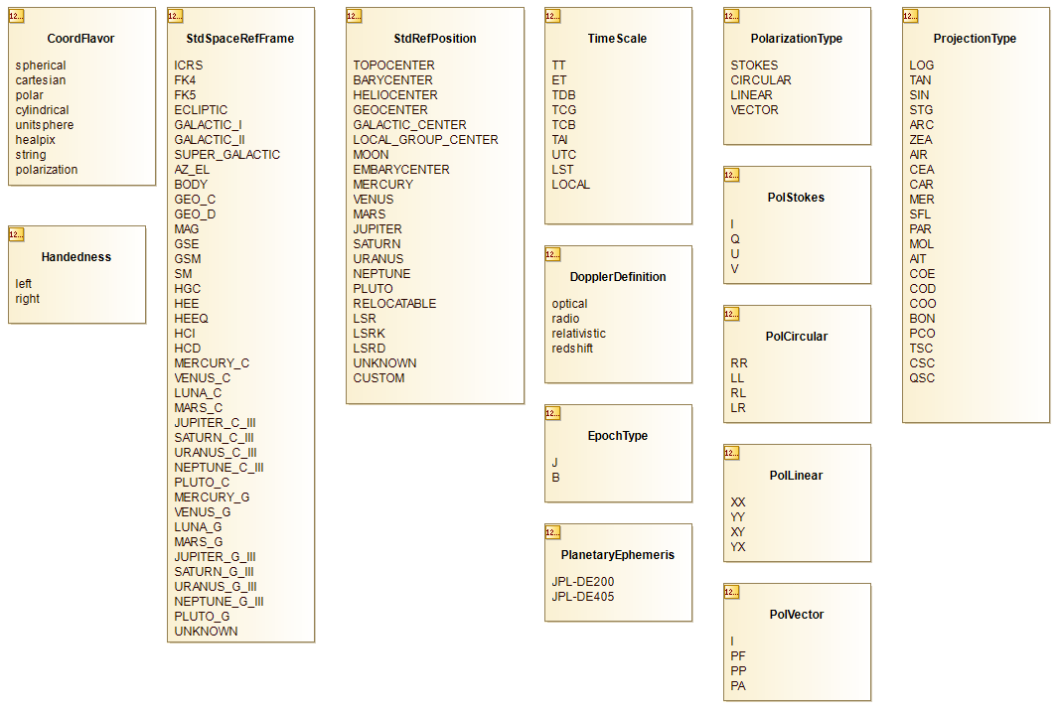


Figure 28: STC Enumerations

6.7.1 CoordFlavor

Token	Meaning
spherical	Spherical 2-D (long,lat) or 3-D (long, lat, rad/elev)
cartesian	Cartesian 1-, 2-, or 3-D coordinates
polar	2-D polar coordinates (radius, posangle)
cylindrical	3-D cylindrical coordinates (radius, posangle, z)
unitsphere	3-D Unit sphere coordinates (direction cosines)
healpix	2-D Healpix coordinates
string	String coordinates
polarization	Polarization coordinates

6.7.2 DopplerDefinition

Token	Meaning
optical	
radio	

relativistic	
redshift	

6.7.3 EpochType

Token	Meaning
J	Julian
B	Besselian

6.7.4 Handedness

Token	Meaning
left	Left handed
right	Right handed

6.7.5 PlanetaryEphemeris

Enumeration of JPL Planetary Ephemerides.

Token	Note
JPL-DE200	typically used with FK5
JPL-DE405	typically used with ICRS

6.7.6 PolarizationType

Enumeration of polarization types.

Token	Meaning
STOKES	
CIRCULAR	
LINEAR	
VECTOR	

6.7.7 PolStokes

Enumeration of Stokes polarization states.

Token	Meaning
I	Standard Stokes unpolarized
Q	Standard Stokes linear
U	Standard Stokes linear
V	Standard Stokes circular

6.7.8 PolCircular

Enumeration of Circular polarization states.

Token	Meaning
RR	Right-right circular
LL	Left-left circular
RL	Right-left cross-circular
LR	Left-right cross-circular

6.7.9 PolLinear

Enumeration of Linear polarization states.

Token	Meaning
XX	X parallel linear
YY	Y parallel linear
XY	XY cross linear
YX	YX cross linear

6.7.10 PolVector

Enumeration of Vector polarization states.

Token	Meaning
I	
PF	
PP	
PA	

6.7.11 ProjectionType

Enumeration of WCS Projection types.

Token	Meaning
LOG	Linear to logarithmic
TAN	Tangent plane projection
SIN	Sine projection
STG	Stereographic projection
ARC	Zenithal equidistant projection
ZEA	Zenithal equal-area projection
AIR	Airy projection
CEA	Cylindrical equal-area projection
CAR	Plate Carre projection
MER	Mercator projection
SFL	Sanson-Flamsteed projection
PAR	Parabolic projection
MOL	Mollweide projection

AIT	Hammer-Aitoff projection
COE	Conic equal-area projection
COD	Conic equidistant projection
COO	Conic orthomorphic projection
BON	Bonne equal-area projection
PCO	Polyconic projection
TSC	Tangential spherical cube projection
CSC	COBE quadrilatealized spherical cube projection
QSC	Quadrilateralized spherical cube projection

6.7.12 StdSpaceRefFrame

Enumeration of standard Space Reference Frames.

Token	Meaning
ICRS	The ICRS frame
FK4	FK4
FK5	FK5
ECLIPTIC	Ecliptic l,b
GALACTIC_I	Old galactic LI, BI
GALACTIC_II	Galactic LII, BII
SUPER_GALACTIC	SGL, SGB
AZ_EL	Azimuth and elevation
BODY	Generic Body (e.g. planet)
GEO_C	Geocentric corotating
GEO_D	Geodetic ref frame
MAG	Geomagnetic ref frame
GSE	Geocentric Solar Ecliptic
GSM	Geocentric Solar Magnetic
SM	Solar Magnetic
HGC	Heliographic
HEE	Heliocentric Earth Ecliptic
HEEQ	Heliocentric Earth Equatorial
HCI	Heliocentric Inertial
HCD	Heliocentric of Date
MERCURY_C	Corotating planetocentric
VENUS_C	Corotating planetocentric
LUNA_C	Corotating planetocentric
MARS_C	Corotating planetocentric
JUPITER_C_III	Corotating planetocentric
SATURN_C_III	Corotating planetocentric
URANUS_C_III	Corotating planetocentric
NEPTUNE_C_III	Corotating planetocentric

PLUTO_C	Corotating planetocentric
MERCURY_G	Corotating planetographic
VENUS_G	Corotating planetographic
LUNA_G	Corotating planetographic
MARS_G	Corotating planetographic
JUPITER_G_III	Corotating planetographic
SATURN_G_III	Corotating planetographic
URANUS_G_III	Corotating planetographic
NEPTUNE_G_III	Corotating planetographic
PLUTO_G	Corotating planetographic
UNKNOWN	Unknown frame
CUSTOM	Custom frame

6.7.13 StdSpaceRefPosition

Enumeration of Standard Reference Positions.

Token	Meaning	Note
TOPOCENTER	Location of the observing device	(telescope)
BARYCENTER	Solar system barycenter	
HELIOCENTER	Center of the Sun	
GEOCENTER	Center of the Earth	
GALACTIC_CENTER	Center of the Galaxy	
LOCAL_GROUP_CENTER	Barycenter of the Local Group	
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	Relative origin	Suitable for simulations
LSR	Local Standard of Rest	Spectral/Redshift domain only
LSRK	Kinematic Local Standard of Rest	Equivalent to LSR
LSRD	Dynamic Local Standard of Rest	Spectral/Redshift domain only
UNKNOWN	Unknown origin	
CUSTOM	Origin specified wrt another system	

6.7.14 TimeScale

Enumeration of time scales.

Token	Meaning
LOCAL	Relocatable (simulation) time
TT	Terrestrial Time
UTC	Coordinated Universal Time
ET	Ephemeris Time
TDB	Barycentric Dynamical Time
TCG	Terrestrial Coordinate Time
TCB	Barycentric Coordinate Time
TAI	International Atomic Time
LST	Local Sidereal Time

6.8 STC Model Base Types

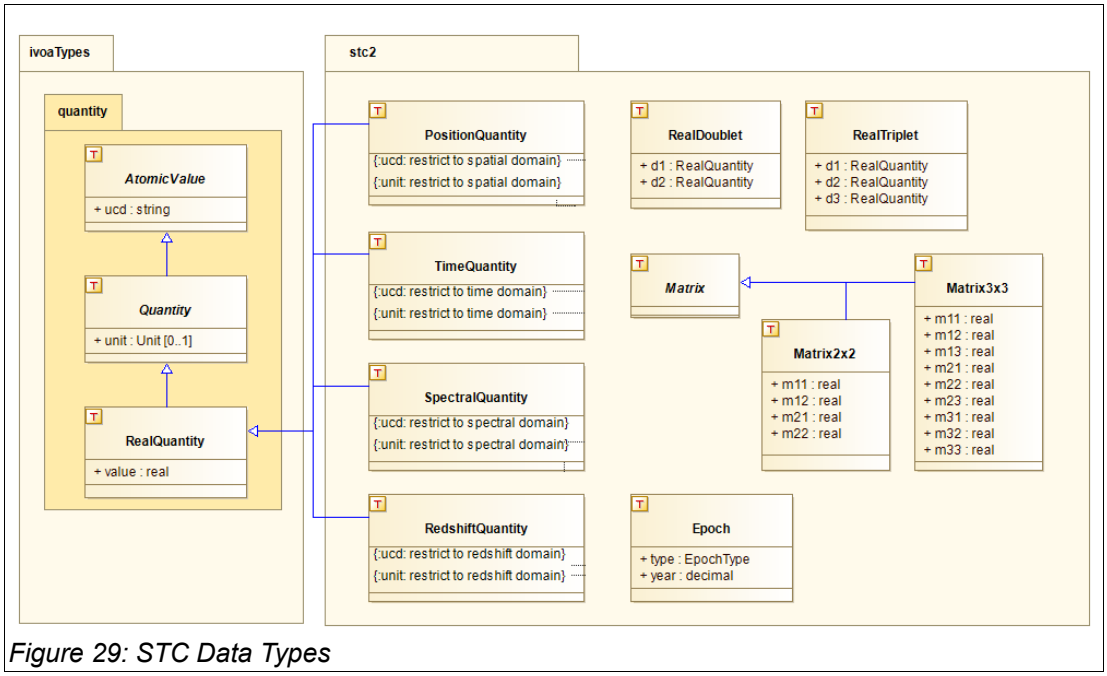


Figure 29: STC Data Types

6.8.1 PositionQuantity

Extension of ivoa::RealQuantity specialized for the Spatial domain. This type enforces that the ucd and unit attributes are appropriate for that domain.

6.8.2 TimeQuantity

Extension of ivoa::RealQuantity specialized for the Time domain. This type enforces that the ucd and unit attributes are appropriate for that domain. NOTE: The value of this type does not include the ivoa::datetime type, so is only appropriate for numerical time representations such as MJD, JD, or mission elapsed time.

6.8.3 SpectralQuantity

Extension of ivoa::RealQuantity specialized for the Spectral domain. This type enforces that the ucd and unit attributes are appropriate for that domain.

6.8.4 RedshiftQuantity

Extension of ivoa::RealQuantity specialized for the Spectral domain. This type enforces that the ucd and unit attributes are appropriate for that domain.

6.8.5 RealDoublet

Type to hold a pair of RealQuantity types. This object is intended to be used when it is important to emphasize the dimensionality of a value pair, rather than using an array representation.

6.8.5.1 d1:RealQuantity

6.8.5.2 d2:RealQuantity

6.8.6 RealTiplet

Type to hold a triplet of RealQuantity types. This object is intended to be used when it is important to emphasize the dimensionality of a value pair, rather than using an array representation.

6.8.6.1 d1:RealQuantity

6.8.6.2 d2:RealQuantity

6.8.6.3 d3:RealQuantity

6.8.7 Matrix

Abstract head of the matrix classes.

6.8.8 Matrix2x2

Basic 2 x 2 matrix. Contains attribute for each matrix element as real type.

6.8.9 Matrix3x3

Basic 3 x 3 matrix. Contains attribute for each matrix element as real type.

6.8.10 Epoch

Type for astronomical epoch date.

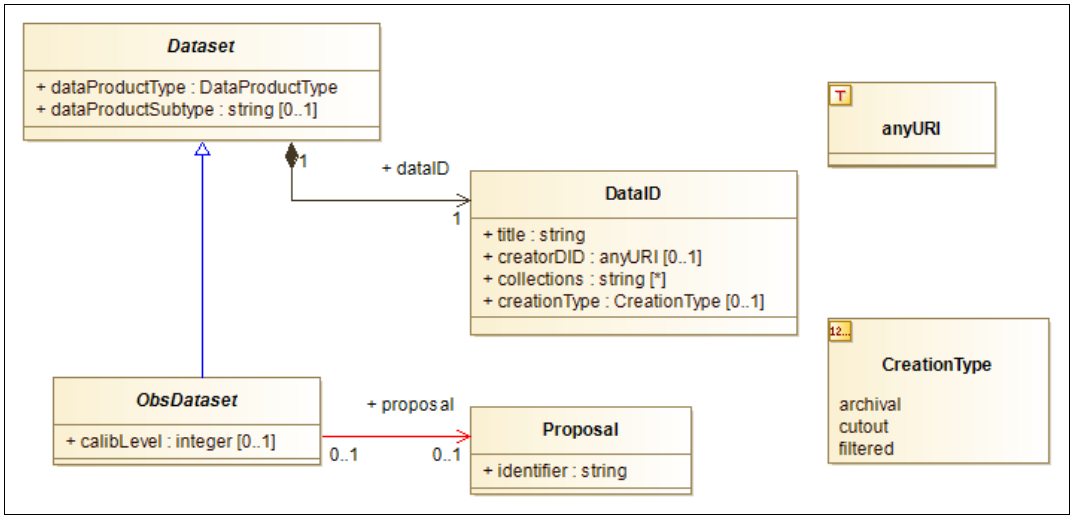
6.8.10.1 type:EpochType

6.8.10.2 year:decimal

Appendix A: Modeling Conventions

1 Diagram notation

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



1.1 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.

1.2 DataType

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

1.3 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.

1.4 Generalization

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

1.5 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 (1) 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.

1.6 Reference

The reference relation is indicated by a red 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.

1.7 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).

2 Model Identification metadata

Interoperability of datasets requires that there be a standardized method for identifying the specific type of dataset, and which model(s) and versions thereof it conforms to. These elements are not properties of the dataset, but rather, of the Model itself. We provide this information via stereotypes assigned to the model packages (e.g. Dataset, Char, STC, IVOA).

2.1 Model stereotype

The Model stereotype (<<model>>) consists of a set of Model properties which identify a particular model and its dependencies. Each model should specifically state the appropriate values for these properties.

2.1.1 name:string[1]

The model name. The value must match the name of the model package itself. This string identifies the particular model type (eg. Dataset, Char, STC).

2.1.2 version:string[1]

The version of this model. To be represented as a string with format "<version>.<subversion>"

2.1.3 prefix:string[1]

Sometimes referred to as 'namespace', the prefix is a tag which is used to label elements of a particular model. Each model must declare a prefix string which is unique within the IVOA to tag

elements from those models. A typical use of the prefix is in the construction of element UTYPE strings.

2.1.4 url:anyURI[1]

A URL from which the full model description may be obtained (e.g. XML schema).

2.1.5 imports:Import[*]

Here, we specify which the models on which this model is dependent. This model uses and/or extends elements from the Characterisation and STC Data models. In this document, we provide descriptions and supporting information about usage of these objects in a particular context. The originating documents, however, remain the definitive source for element definitions.

2.2 Import Stereotype

The <<import>> stereotype is attached to Packages representing imported models. It identifies the model by name, and provides URLs from which the full description may be obtained.

2.2.1 name:string[1]

The name of the imported model. This name MUST match the 'name' property of imported model's Model metadata.

2.2.2 version:string[1]

The version of this model. To be represented as a string with format "<version>.<subversion>"

2.2.3 url:anyURI[1]

A URL from which the full model description may be obtained (e.g. XML schema).

3 Extensibility

There is no formal mechanism in the IVOA defining how users may extend models with their own content. However, the above Model identification metadata provides a simple means to do so. Using this process, a user would model their content as an extension of the IVOA standard.

3.1 Model

3.1.1 name

The user-defined model would need a name unique from that of the standard.

3.1.2 prefix

A unique prefix must be defined for the user-defined model elements. Users must take care not to make use of prefix tags which are associated with current IVOA standards, (e.g. ’cha’, ’spec’, ’ssa’, ’stc’). At the time of this writing, there is no central repository of reserved namespace strings.

3.1.3 imports

The user defined model should declare the IVOA standard being extended as an imported model. Fields for the imported model name and url may be obtained from that standard's documentation.

3.2 Scope

We permit any object modeled in this document to be extended with user-defined content, with the following restrictions:

- Follow VO-DML modeling practices.
- Values of extended content must be consistent with the content of modeled data. That is, using the IVOA base primitive types, Quantity, and STC Coordinates as appropriate.
- Since extended content, by definition, does not follow the corresponding model, it is not possible for general applications to interpret complex structures within that content. It is, therefore, recommended that users define extended content in such a way as to avoid ambiguity between its components.

3.3 Support

Applications should, but are not required to, provide the following support for extended content:

- Retain existence of extended content, including namespace and UTypes.
- Retain association with modeled component.
- Provide access to extended content by users.

Appendix C: Dataset Metadata Model Summary

Dataset Model Identification				
Model Element	**Datatype**	**Mult.**	**Meaning**	**value**
Model identification				
Model				
Model.name	string	1	Data model name and version	"Dataset Metadata"
Model.version	string	1	Data model version	"1.0"
Model.prefix	string	1	Data model prefix tag	"ds"
Model.url	anyURI	1	Reference URL for model	<TBD>
Imported Model				
Import.name	string	1	Imported model name	"char"
Import.version	string	1	Imported model version	"1.13"
Import.url	anyURI	1	Reference URL for imported model	<TBD>
Imported Model				
Import.name	string	1	Imported model name	"stc2"
Import.version	string	1	Imported model version	"2.0"
Import.url	anyURI	1	Reference URL for imported model	<TBD>
Imported Model				
Import.name	string	1	Imported model name	"ivoa"
Import.version	string	1	Imported model version	"1.0"
Import.url	anyURI	1	Reference URL for imported model	<TBD>

Dataset Model Summary				
Model Element	**Datatype**	**Mult.**	**Meaning**	**UCD1+**
Dataset Model Elements				
Characterisation			Direct extension of Chararcterisation:Char	
Contact				
Contact.name	string	1	Contact name	meta.bib.author;meta.curation
Contact.email	string	0..1	Contact email	meta.ref.url;meta.email
Curation				
Curation.publisher	string	1	Publisher	meta.curation
Curation.publisherID	anyURI	0..1	URI for VO Publisher	meta.ref.url;meta.curation

Dataset Model Summary				
Model Element	**Datatype**	**Mult.**	**Meaning**	**UCD1+**
Curation.publisherDID	string	0..1	Publisher specified dataset ID	meta.ref.url;meta.curation
Curation.version	string	0..1	Publisher version of the dataset	meta.version;meta.curation
Curation.rights	RightsType	0..1	Proprietary restrictions level	meta.code
Curation.releaseDate	datetime	0..1	Date curated dataset last modified	time.release
Curation.references	string	0..*	URL or Bibcode for documentation	meta.bib.bibcode
Curation.contact	Contact	0..1		
DataID				
DataID.title	string	1	Dataset title	meta.title;meta.dataset
DataID.datasetID	anyURI	0..1	IVOA Dataset Identifier	meta.id;meta.dataset
DataID.creatorDID	anyURI	0..1	Creator defined Dataset Identifier	meta.id
DataID.date	datetime	0..1	Data processing/creation date	time.epoch;meta.dataset
DataID.creator	string	0..1	VO Creator ID	meta.curation
DataID.collections	string	0..*	Collection name(s)	meta.id
DataID.version	string	0..1	Version of dataset	meta.version;meta.dataset
DataID.creationType	CreationType	0..1	Dataset creation type	
DataID.logo	anyURI	0..1	URL for creator logo	meta.ref.url
DataID.contributors	string	0..*	Contributor(s)	
DataID.observationID	string	0..1	Observation ID	meta.id
Dataset				
Dataset.dataProductType	DataProductType	1	Dataset or segment type	meta.id
Dataset.dataProductSubType	string	0..1	Dataset subtype	meta.id
Dataset.curation	Curation	1	Dataset curation metadata	
Dataset.dataID	DataID	1	Dataset identification metadata	
Derived				
Derived.snr	real	0..1	Signal-to-noise ratio	stat.snr
Derived.varAmpl	real	0..1	Variability amplitude as fraction of mean	src.var.amplitude;arith.ratio
Derived.redshift	Redshift	0..1	Measured redshift	
ObsDataset				
ObsDataset.calibLevel	integer	0..1	Calibration level	meta.code;obs.calib
ObsDataset.characterisation	Characterisation	1		
ObsDataset.coordSys	CoordSys	0..*	Global coordinate systems	
ObsDataset.derived	Derived	0..1	Derived metadata	
ObsDataset.obsConfig	ObsConfig	0..1	Observatino configuration	
ObsDataset.proposal	Proposal	0..1	Proposal information	
ObsDataset.target	BaseTarget	1		
Redshift			Extension of STC:RedshiftCoord	
Redshift.coord	Quantity	1	Measured redshift value	src.redshift
Redshift.StatError	Uncertainty	0..1	Error on measured redshift	stat.error;src.redshift

Dataset Model Summary				
Model Element	**Datatype**	**Mult.**	**Meaning**	**UCD1+**
Redshift.Confidence	real		Confidence value on redshift	
Observation Model Elements				
AstroTarget			Astronomical target	
AstroTarget.objectClass	string	0..1	Target or object class	src.class
AstroTarget.spectralClass	string	0..1	Object spectral class	src.spType
AstroTarget.redshift	real	0..1	Target redshift	src.redshift
AstroTarget.VarAmpl	real	0..1	Target variability amplitude - typical	src.var.amplitude
BaseTarget				
BaseTarget.name	string	1	Target name	meta.id;src
BaseTarget.description	string	0..1	Target descriptive text	meta.note;src
BaseTarget.position	Position	0..1	Target location (eg: RA, DEC)	pos[.eq];src
Target			Generic Target	
Target.objectClass	string	0..1	Target or object class	src.class
ObsConfig				
ObsConfig.observingElement	ObservingElement	0..*	Observation configuration parameter	
ObservingElement				
ObservingElement.name	string	1	Identifies the specific instance of the element	
Bandpass			Direct extension of ObservingElement	
Bandpass.name	string	1	Band	instr.bandpass
DataSource			Direct extension of ObservingElement	
DataSource.name	string	1	Original data type	
Facility			Direct extension of ObservingElement	
Facility.name	string	1	Facility name	meta.id;instr.tel
Instrument			Direct extension of ObservingElement	
Instrument.name	string	1	Instrument ID	meta.id;instr
Observation				
Observation.observationID	string	1	Observation ID	
Observation.obsConfig	ObsConfig	1	Observation configuration metadata	
Observation.proposal	Proposal	0..1	Proposal which spawned the observation	
Observation.result	ObsDataset	0..*	Dataset(s) resulting from the observation	
Observation.target	BaseTarget	1	Target or goal of the observation	
Proposal				
Proposal.Identifier	string	1	Proposal ID	meta.id;obs.proposal

References

[1] *Resource Metadata for the Virtual Observatory*: Version 1.12, 02 March 2007
http://www.ivoa.net/Documents/latest/RM.html
[2] ”IVOA Identifiers”: Version 1.12, 14 March 2007
http://www.ivoa.net/Documents/latest/IDs.html
[3] ”Data Model for Astronomical DataSet Characterisation”: Version 1.13
http://www.ivoa.net/Documents/REC/DM/CharacterisationDM-20080325.pdf
[4] ”ST-ECF newsletter, issue #42”, June 2007
http://www.spacetelescope.org/about/further information/newsletters/html/newsletter42.html
[5] ”DER SNR: A simple and general spectroscopic signal-to-noise measurement algorithm”;
http://www.stecf.org/software/ASTROsoft/DER SNR
[6] ”Simulation Data Model”: Version 1.0, 03 May 2012;
http://www.ivoa.net/documents/SimDM/20120503/REC-SimulationDataModel-1.00-20120503.pdf
[7] ”An encoding system to represent stellar spectral classes in archival databases and catalogs": Version 1.04, 2011 Dec 15
http://www.ivoa.net/documents/Notes/SpectClasses/20111215/SpectClasses-20111215.pdf
[8] ”VO-DML a consistent modeling language for IVOA data models”: Version 1.00-20140427
http://volute.googlecode.com/svn/trunk/projects/dm/vo-dml/doc/VO-DML-WD-v1.0.pdf
[9] ”Units in the VO”: Version 1.0
http://www.ivoa.net/Documents/VOUnits/20120820/PR-VOUnits-1.0-20120820.pdf
[10] ”The UCD1+ controlled vocabulary”: Version 1.23
http://www.ivoa.net/Documents/REC/UCD/UCDlist-20070402.pdf
http://cdsweb.u-strasbg.fr/UCD/ucd1p-words.txt
[11] ”An IVOA Standard for Unified Content Descriptors”: Version 1.10
http://www.ivoa.net/Documents/REC/UCD/UCD-20050812.pdf
http://www.ivoa.net/Documents/latest/UCD.html