



*International
Virtual
Observatory
Alliance*

IVOA N-Dimensional Cube Model

Version 1.0

IVOA Working Draft 20150320

This version:

[WD-CubeDM-1.0-20140930](#)

Previous version(s):

Editor(s):

Mark Cresitello-Dittmar

Authors:

Doug Tody, Francois Bonnarel, Omar Laurino, Mireille Louys, Arnold Rots, Jose Enrique Ruiz, Jesus Salgado, and the IVOA Data Model Working Group.

Abstract

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:

2013 May 05: Initial release of ImageDM draft document.

2013 Aug 16: Rev 16 updates.

2013 Nov 15: Second draft of ImageDM release.

Updated to reflect initial reviewer comments. Architecture section extensively revised to provide compatibility with ObsCore/Observation and Char2

2014 Sep 10: Model restructured as extension of Observation Dataset

Extensive revision, basing Cube model as extension of dependent models: Dataset, and STC

Define new components defining generic Cube datasets (image and sparse) which can be utilized for Spectral and other models

Baseline to STC-2.0 prototype model.

2015 Mar 20: Updated STC-2.0 prototype model spec.

Contents

1	Introduction.....	6
1.1	Motivation.....	6
1.2	Use Cases.....	6
1.3	Cube Classifications.....	7
1.4	IVOA Architecture Context.....	9
1.5	Model Dependencies.....	9
1.6	Structure of this Documentation.....	10
2	N-Dimensional Datasets.....	11
2.1	NDImageDataset.....	11
2.1.1	NDImageDataset.image.....	12
2.2	Sparse Cube Dataset.....	13
2.2.1	SparseCubeDataset.cube.....	13
3	N-Dimensional Data Products.....	14
3.1	DataProduct.....	14
3.1.1	DataProduct.coordSys.....	14
3.1.2	DataProduct.mappings.....	14
3.2	PixelatedData.....	15
3.2.1	PixelatedData.pixelCoordSys.....	15
3.3	PointData.....	15
3.3.1	PointData.customAxes.....	15
3.4	CoordSys.....	15
3.5	FrameMappings.....	15
3.6	PixelCoordSys.....	15
4	N-Dimensional Image.....	16
4.1	NDImage.....	16
4.1.1	NDImage.data.....	17
4.2	Voxel.....	17
4.2.1	Voxel.pixelAxes.....	17
4.2.2	Voxel.dependentAxis.....	17
5	Sparse Cube.....	18
5.1	SparseCube.....	18
5.1.1	SparseCube.data.....	18
5.2	NDPoint.....	19
5.2.1	NDPoint.spaceAxis.....	19
5.2.2	NDPoint.timeAxis.....	19
5.2.3	NDPoint.spectralAxis.....	19
5.2.4	NDPoint.redshiftAxis.....	19
5.2.5	NDPoint.polarizationAxis.....	19
6	Data Types.....	20
6.1	IVOA Data Types.....	20
6.1.1	Units.....	20
6.1.2	UCDs.....	21

6.1.3Dates.....	21
6.2Cube Model DataTypes.....	21
Appendix A: Modeling Conventions.....	22
1Diagram notation.....	22
2Model Identification metadata.....	22
2.1Model stereotype.....	22
2.2Import Stereotype.....	22
3Extensibility.....	22
Appendix B: N-Dimensional Cube Model Summary.....	23
References.....	25

1 Introduction

1.1 Motivation

The concept of the *astronomical image* goes back decades, most notably to the introduction of the flexible image transport system (FITS) in the late 1970s by Wells and Greisen. Later papers by Greisen, Calabretta, and others added the capability to define a *world coordinate system* (WCS) that can be associated with an image to define the physical coordinates in an M-dimensional space of each data sample (observable) in the N-dimensional array comprising the data segment of the image. Other metadata elements (FITS keywords) are defined to describe the origins and content of the particular image dataset. The astronomical multi-dimensional image concept is a type of data hypercube, and is related to the volumetric cubes used in medical and geological applications, and to commercial technologies such as the OLAP cube, which projects a RDBMS relation onto the axes of a hypercube.

An important aspect of the astronomical image is *abstraction*. The image model hides how the data are physically stored; this is especially important for large images or image cubes, which may be Gigabytes or Terabytes in size for a single dataset. While logically the data portion of the image may be a simple N-d array, physically the data may be represented or stored in many different ways. Large cubes may be physically stored in multiple smaller segments, or data may be stored in N-d blocks to provide uniform access along any dimension or axis of the image. Sparse cubes may be stored as multiple segments, each at a given location within the larger logical cube. Data may be stored in a compressed form, or may be encoded, e.g., via a multiresolution technique such as a wavelet transform (JPEG2000). Each such representation offers certain advantages and disadvantages; by separating the logical view of the data from the details of how it is physically represented, the optimum choice may be made for each application, transparently to higher-level analysis software.

This model presents an abstracted representation of N-Dimensional cube datasets and serves as a framework on which to construct models for more specialized Astronomical datasets.

1.2 Use Cases

A use case study of multidimensional/cube data performed by the US VAO and community partners in early 2013 contributed to early development of this data model. A special session on multidimensional data held by the IVOA in May 2013 helped to further refine community use cases for the data model.

A comprehensive analysis of use cases for the Image data model in the context of multidimensional astronomical data is beyond the scope of this document, but is available in **[TBD]**: *reference to VAO use case analysis*].

Here we summarize major use cases from this study:

- Simple 2-D image, the most common astronomical image type.
- A single cube image is often sparse in either the spectral or spatial or temporal plane. The JCMT cubes are a good example of a spectral data cube that is sparse in the spatial plane.
- Multiple-subarray images are common for instrumental observations composed of multiple detectors. For example, a spectral data cube with multiple spectral bands, or a

CCD mosaic where the detectors may not be exactly aligned, requiring different WCS calibrations.

- Very large cubes are increasingly common with modern instruments. It becomes physically impossible to store these as a single cube dataset (file); large cubes must be stored as multiple files, often with sophisticated voxel encoding, compression, tiling, or other data representation algorithms. The logical image abstraction needs to hide this complexity from client applications to simplify and unify data access.
- A key use case for using image/cube data is interactive image visualization and analysis, particularly for very large cubes. In this case, the image dataset is initially rendered in 2-D, greatly reduced in size, then the user interactively slices and dices the cube, computes 2-D projections, extracts spectra, computes moments, etc., to interactively view the contents of the cube. For very large cubes, this must be done via remote access to the cube data, which is staged to a parallel cluster providing the parallel computing capability required to quickly render the large cube, that may be tens to hundreds of GB in size or larger.

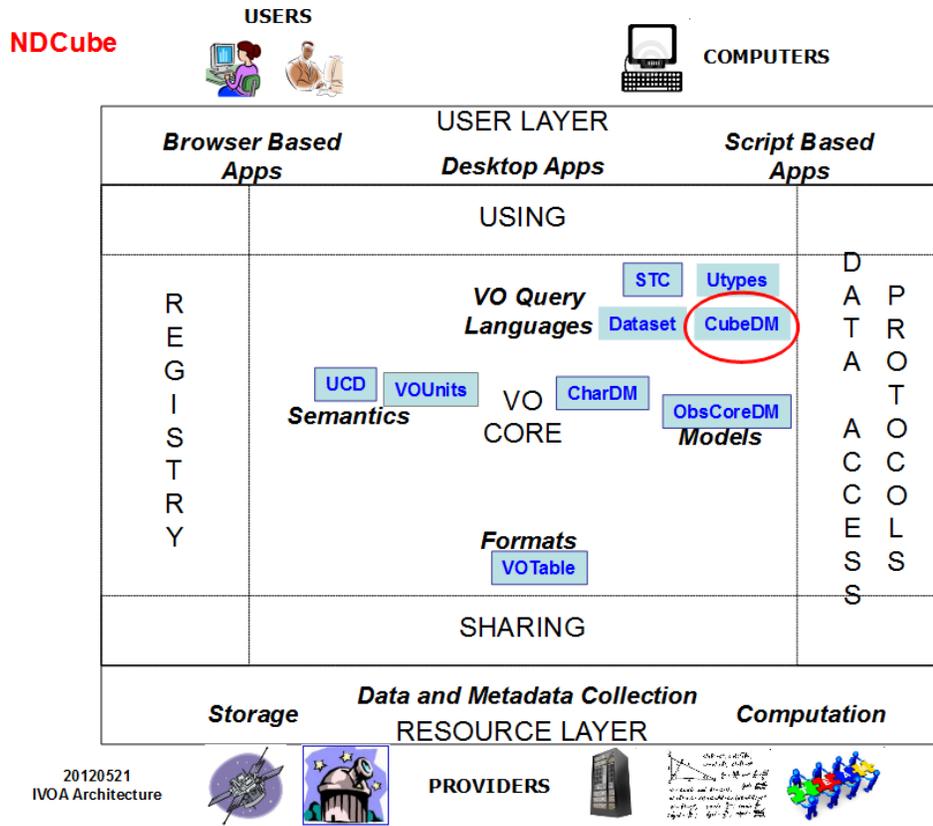
1.3 Cube Classifications

From the primary use cases, we can define these classifications of cube data to be addressed by this model:

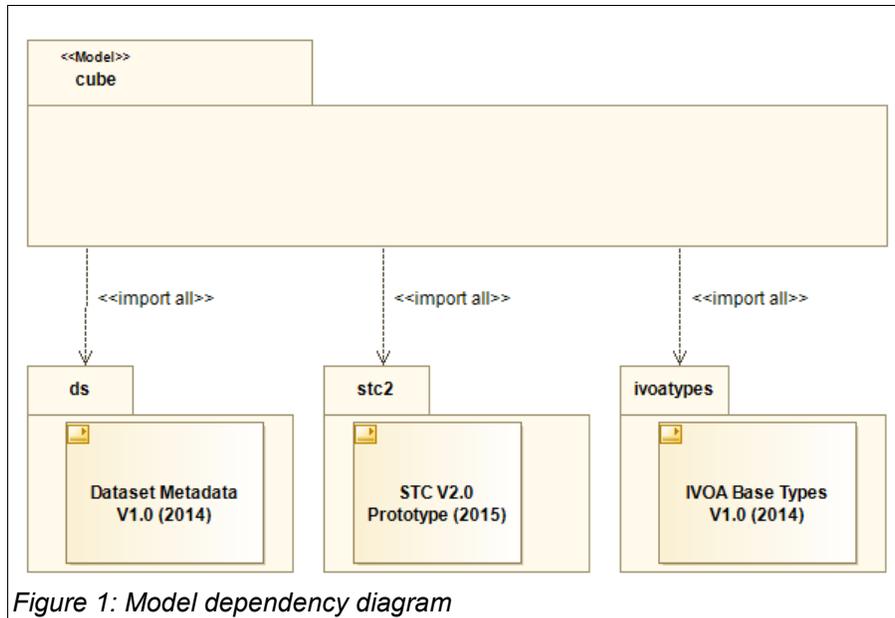
1. **Simple image.** A single filled or mostly filled N-Dimensional image array with associated metadata describing the dataset, its WCS coordinate systems, etc.
2. **Simple sparse image.** A single N-Dimensional array as in case #1, however, large portions of the image may be sparse (have no data samples).
3. **Multiple subarray image.** An image dataset containing multiple subarrays, i.e., N-Dimensional image data arrays within the coverage of the overall image dataset. The subarrays may differ in size, resolution, coverage, or other characteristics. The overall image dataset containing the subarrays may be sparse. The overall image dataset does not have an explicit image geometry or sampling, only coverage. There are two sub-cases here:
 - a) the subarrays are all part of the same observation and share common metadata. For example, a multi-band image observation.
 - b) the subarrays may differ arbitrarily and the overall image dataset is essentially an aggregation of (possibly sparse) simple images as in case #1 or #2. For example, a complex image aggregating data from multiple observations.
4. **Large cube.** Very large cubes may be represented as either a simple image or multi-subarray type. The cube model abstraction does not impose any size limitations. The important factor here, is that the chosen view of the cube is independent of the physical storage structure.
5. **Wide-field survey.** This is essentially the same as #3, except that no subarrays (no explicitly pixelated data) or individual image datasets may be externally visible. The survey has coverage, but only automated virtual data generation techniques may be used to access the data, with sub-regions being computed on-the-fly and returned to the client. Alternatively, a survey might expose a collection of discrete cubes, in which case #1 or #3 may be used.
6. **Sparse data cube.** Sparse data are commonly used for higher-dimensional cubes, and are frequently sparse along one or more axes. For example, a multi-band image has

data at only a few given spectral coordinates, (each corresponding to a spectral bandpass). A spectral (or velocity) data cube may contain data for a number of widely spaced spectral bands, each of which may differ in the spectral resolution and number of channels. A time cube likewise may contain data, either individual points, or time series, arbitrarily spaced along the time axis with time regions where no data was taken. A multi-object spectral data cube may be sparse in the spatial plane. Event data can be considered a data cube which is sparse in all measurement axes.

1.4 IVOA Architecture Context



1.5 Model Dependencies



The ND-Cube model is built on other IVOA data models as indicated in Figure 1. The <<Model>> and <<import>> stereotypes provide information identifying the model, its 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.

1.6 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, the type of the target of the relation, and a reference to the full definition of that type.

2 N-Dimensional Datasets

2.1 NDIImageDataset

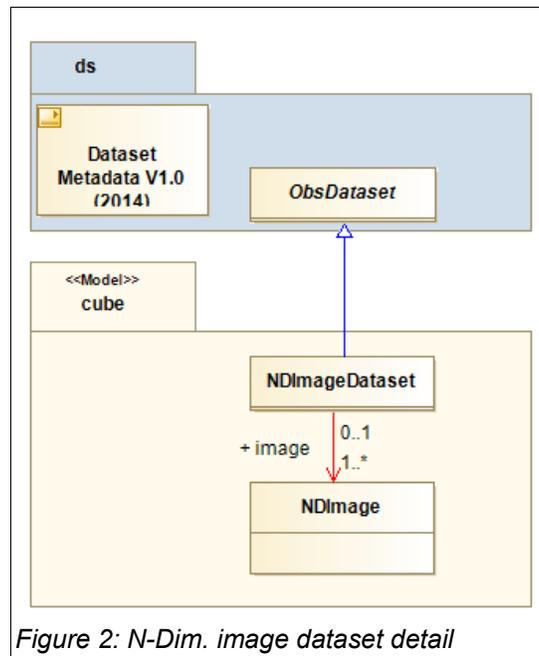


Figure 2: N-Dim. image dataset detail

The NDIImageDataset is an extension of the Observation-Dataset (ObsDataset) object of the Dataset Metadata model which provides all the generic IVOA Dataset metadata (DataID, Curation, etc.), plus metadata related to observational data (Target, Characterisation, ObsConfig, etc.). See sections 2 and 3 of the Dataset Metadata document for a full description of the inherited content.

The IVOA Dataset is defined as one or more files which are considered to be a single deliverable. The NDIImageDataset, should be considered a collection of one or more NDIImage objects, each of which defines a particular image instance.

In terms of our image cube classifications:

- **Simple Image** - a single NDIImageDataset with identifying metadata, and a single NDIImage instance containing the data array definitions, coordinate systems, and (WCS) mapping definitions.
- **Single Sparse Image** - is the same as Simple Image, except in the mapping definitions used to transform the pixel space to world coordinates.
- **Multiple Subarray Image** - Use case 3a consists of a single NDIImageDataset with multiple subarrays, each of which is a separate NDIImage instance. The dataset instance contains metadata for the entire image dataset. Each NDIImage instance contains an N-Dimensional image array with associated metadata, including coordinate mappings, related to that instance. Use case 3b is an aggregation of related but otherwise independent Image instances. This is a form of *complex dataset* wherein standard

datasets are combined to model more complex data. How the individual datasets are related is defined by the application and is not within the scope of this model.

2.1.1 NDImageDataset.image

type: NDImage

type-detail: Section 4.1

multiplicity: 1..*

Reference to one or more NDImage instances. Each instance represents an individual image array, complete with metadata describing its pixel space, coordinate systems, and the mappings between coordinate frames.

2.2 Sparse Cube Dataset

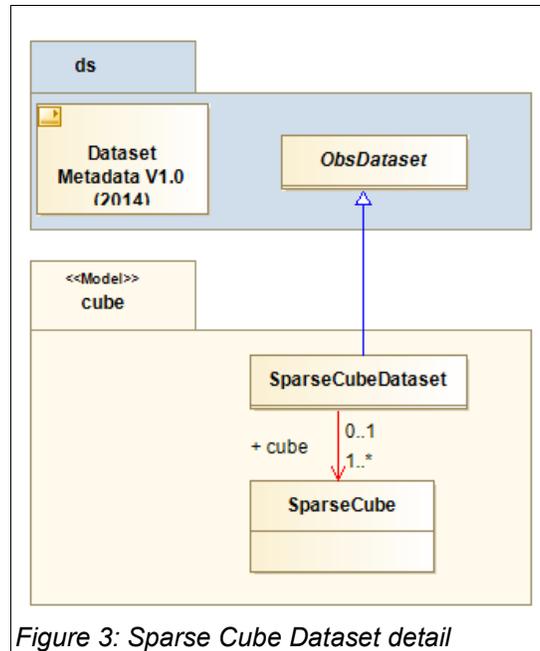


Figure 3: Sparse Cube Dataset detail

The SparseCubeDataset is an extension of the Observation-Dataset (ObsDataset) object of the Dataset Metadata model which provides all the generic IVOA Dataset metadata (DataID, Curation, etc.), plus metadata related to observational data (Target, Characterisation, ObsConfig, etc.). See sections 2 and 3 of the Dataset Metadata document for a full description of the inherited content.

The SparseCubeDataset is suitable for encapsulating N-Dimensional data which are sparse in nature, and do not have a set of regularly sampled (pixel) axes. For example, an event list.

2.2.1 SparseCubeDataset.cube

type: SparseCube
multiplicity: 1..*

type-detail: Section 5.1

Reference to one or more SparseCube instances. Each instance represents an individual cube, complete with metadata describing its coordinate systems, and the mappings between coordinate frames.

3 N-Dimensional Data Products

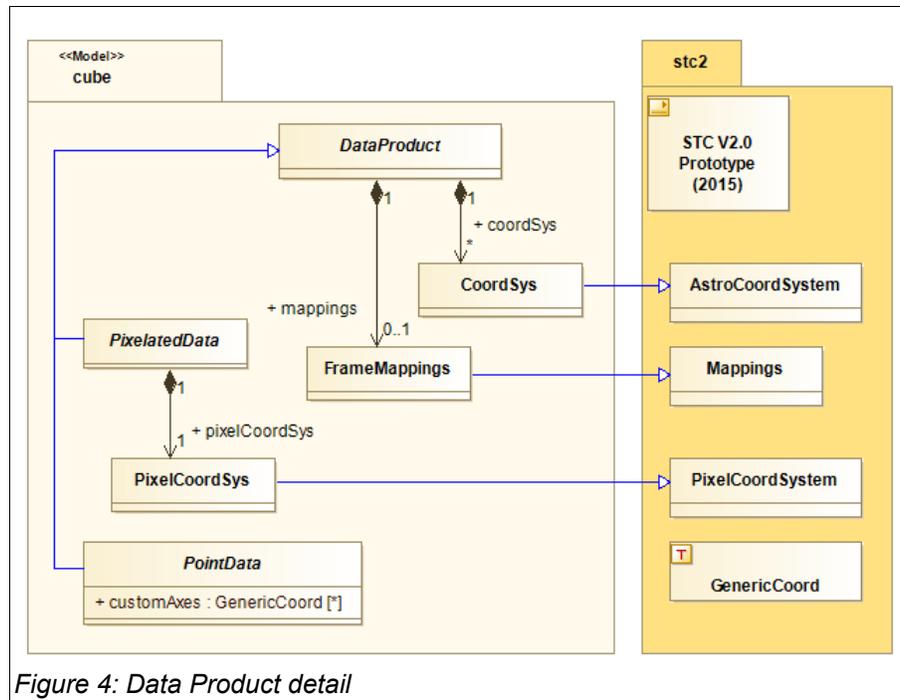


Figure 4: Data Product detail

Data Products represent a single instance of a particular type of data. It contains the minimum set of metadata required to define that type of data. The Data product definition is constant, regardless of what entity or process is generating it. For example, a Spectrum data product has the same structure and content regardless of whether it is a result of an observation, simulation, or fit to a theoretical model.

3.1 DataProduct

Abstract head of the data product tree. All astronomical data products are expected to define the set of coordinate systems associated with the data, and any mappings which define the relations between coordinate frames. Typically mappings are used to define transformations between a pixelated coordinate frame and their corresponding world coordinate frame, but this is not always the case. For example, in sparse cube data such as event lists, mappings may be defined between physical frames (chip or sky) to world coordinate frames (csc or eqpos).

3.1.1 DataProduct.coordSys

type: CoordSys
multiplicity: 0..*

type-detail: Section 3.4

One or more coordinate systems associated with a particular data product.

3.1.2 DataProduct.mappings

type: FrameMappings

type-detail: Section 3.5

multiplicity: 0..1

The FrameMappings object holds a collection of frame transform definitions which facilitate the conversion of coordinates and other objects from one frame representation to another. For example, convert pixel coordinates from the pixel space to world coordinates in a 2D space frame.

3.2 PixelatedData

Abstract extension of DataProduct specialized for pixelated (image) data. In addition to world coordinate systems, a pixelated data product must also define the pixel coordinate system and the pixel space which the image occupies.

3.2.1 PixelatedData.pixelCoordSys

type: PixelCoordSys

type-detail: Section 3.6

multiplicity: 1

The PixelCoordSys object holds the description of the pixel frames, and axis groupings. It also contains the PixelSpace object, which defines the number and span of each pixel axis.

3.3 PointData

Abstract extension of DataProduct specialized for point data.

3.3.1 PointData.customAxes

type: GenericCoord

type-detail: DS Section 6.2.2

multiplicity: 1

Zero or more coordinates associated with frames not represented by a domain specific coordinate type.

3.4 CoordSys

The CoordSys object defined in the cube model is a simple extension of the STC2 AstroCoordSystem object. It holds a collection of Frames which define the N-Dimensional space occupied by the data. The full description of the STC2 AstroCoordSystem can be found in the Dataset Metadata model, section 6.1.2.

3.5 FrameMappings

The FrameMappings object defined in this model is a simple extension of the STC2 Mappings object. This object is a simple container to manage Frame transform definitions which are then available for use by other objects. The full description of STC2 Mappings and Frame transforms can be found in the Dataset Metadata model, sections 6.10 and 6.11.

3.6 PixelCoordSys

The PixelCoordSys object defined in this model is a simple extension of the STC2 PixelCoordSystem object. It holds both a collection of pixel frame definitions for the image, and a PixelSpace object, which defines the number and span of each pixel axis. The full description of the STC2 PixelCoordSystem can be found in the Dataset Metadata model, section 6.8.

4 N-Dimensional Image

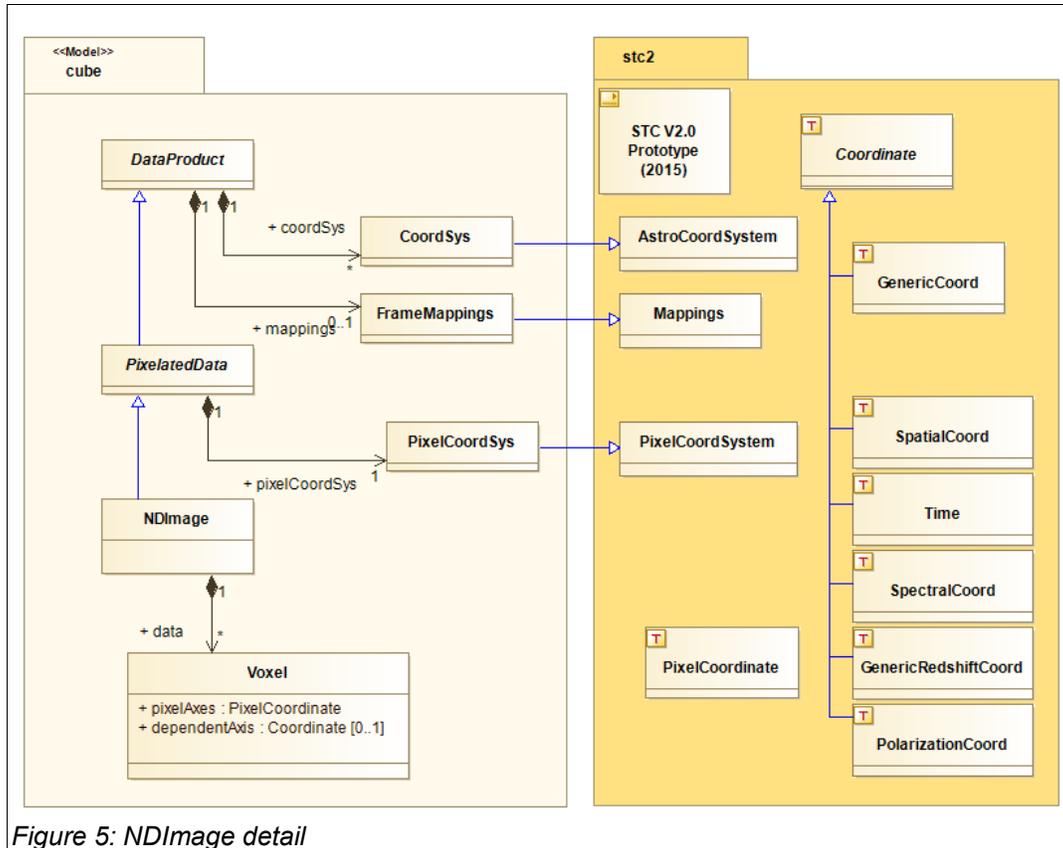


Figure 5: NDIImage detail

4.1 NDIImage

The essential characteristic of an "image" dataset is the presence of a multidimensional, regularly sampled numerical array, with associated metadata describing the image instance. While the concept of an image as a multidimensional array is fully general, astronomical datasets have some combination of specific "domain" related axes (spatial, spectral, time, polarization, etc), with flux or some other derived value, as the array element (voxel) value. The mapping of image pixel axes to physical coordinate frames is described by a set of transformation definitions associated with the image.

Unless dimensionality is otherwise indicated, the terms *image*, *cube*, and *hypercube* are interchangeable and refer to image (array-valued) data of arbitrary dimension. Image is a specialized case of general *hypercube* or *n-cube* data where the value at a given point in the hypercube is restricted to a simple numerical value. The data samples of an image are referred to as *pixels* (picture elements) or as *voxels* (volume elements), pixels being the preferred term for 2D images.

4.1.1 NDImage.data

type: Voxel

type-detail: Section 4.2

multiplicity: 0..*

The NDImage contains a set of zero or more voxels, holding the pixel coordinates in N-D space, and the associated value at that location.

4.2 Voxel

The 'voxel' or volume element, is a fundamental atom on an N-Dimensional image. It is defined by a pixel coordinate, locating the volume in pixel space, and the value at that location.

4.2.1 Voxel.pixelAxes

type: PixelCoordinate

type-detail: DS Section 6.8.5

multiplicity: 1

One or more STC2 PixelCoordinate objects which locates the voxel in N-D pixel space.

4.2.2 Voxel.dependentAxis

type: Coordinate

type-detail: DS Section 6.2.1

multiplicity: 0..1

The 'value' of the Voxel is provided by zero or one STC2 Coordinate objects.

5 Sparse Cube

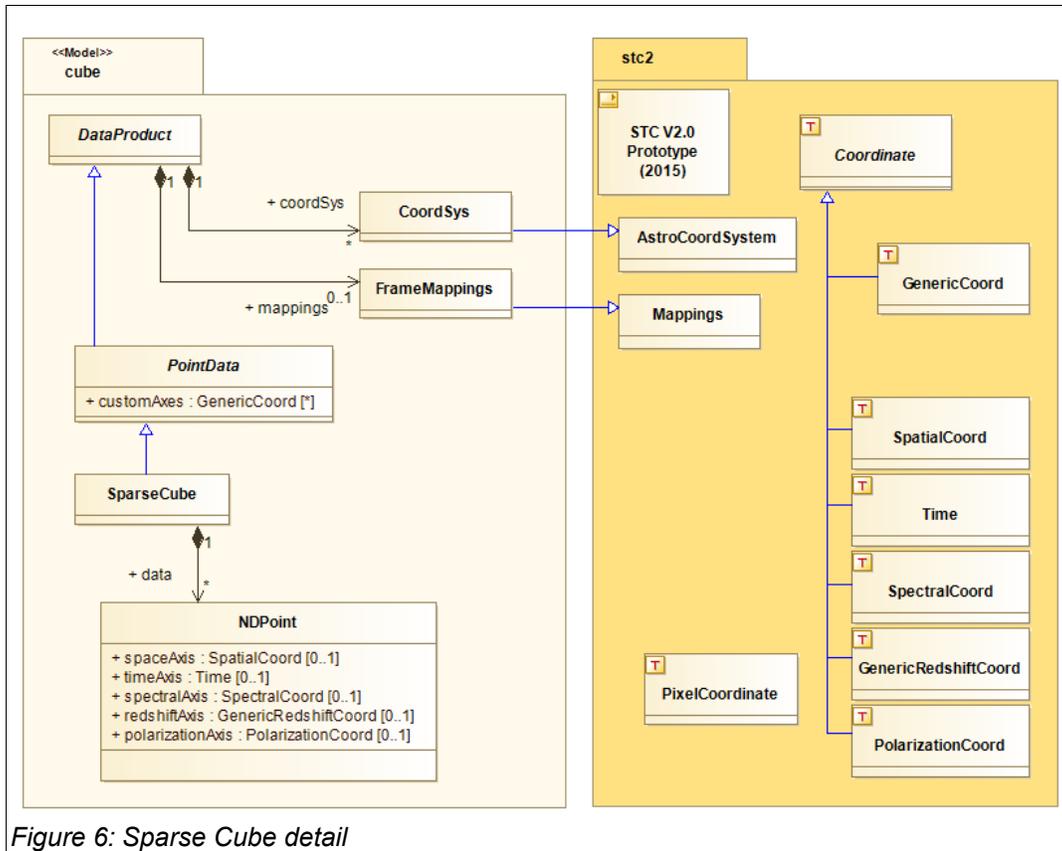


Figure 6: Sparse Cube detail

5.1 SparseCube

An extension of **PointData**, the **SparseCube** is a collection of N-Dimensional points, each of which provide a value for the point in each represented domain. Any domains which are not represented by the specialized domain types, can provide values through the **customAxes** element.

5.1.1 SparseCube.data

type: **NPoint**
multiplicity: 0..*

type-detail: Section 5.2

Zero or more **NPoints**, providing the sparsely sampled data values in each represented domain.

5.2 NDPoint

The NDPoint object is defined by collection of values in each of the defined astronomical domains. As a unit, these completely define a Point in N-Dimensional coordinate space and it is required that each coordinate in the ensemble are represented in frames which comprise the same coordinate system.

5.2.1 NDPoint.spaceAxis

type: SpatialCoord **type-detail:** DS Section 6.4.1
multiplicity: 0..1

Zero or one or more STC2 SpatialCoord objects which provide the spatial location of the point.

5.2.2 NDPoint.timeAxis

type: Time **type-detail:** DS Section 6.3.1
multiplicity: 0..1

Zero or one or more STC2 Time objects which provide the location of the point in the temporal domain. For example, a photon event time.

5.2.3 NDPoint.spectralAxis

type: SpectralCoord **type-detail:** DS Section 6.5.1
multiplicity: 0..1

Zero or one or more STC2 SpectralCoord objects which provide the location of the point in the spectral domain.

5.2.4 NDPoint.redshiftAxis

type: GenericRedshiftCoord **type-detail:** Section 6.6.1
multiplicity: 0..1

Zero or one or more STC2 GenericRedshiftCoord objects which provide the value for the point in the redshift domain.

5.2.5 NDPoint.polarizationAxis

type: PolarizationCoord **type-detail:** Section 6.7.1
multiplicity: 0..1

Zero or one or more STC2 PolarizationCoord objects which provide the polarization state of the point.

6 Data Types

6.1 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.

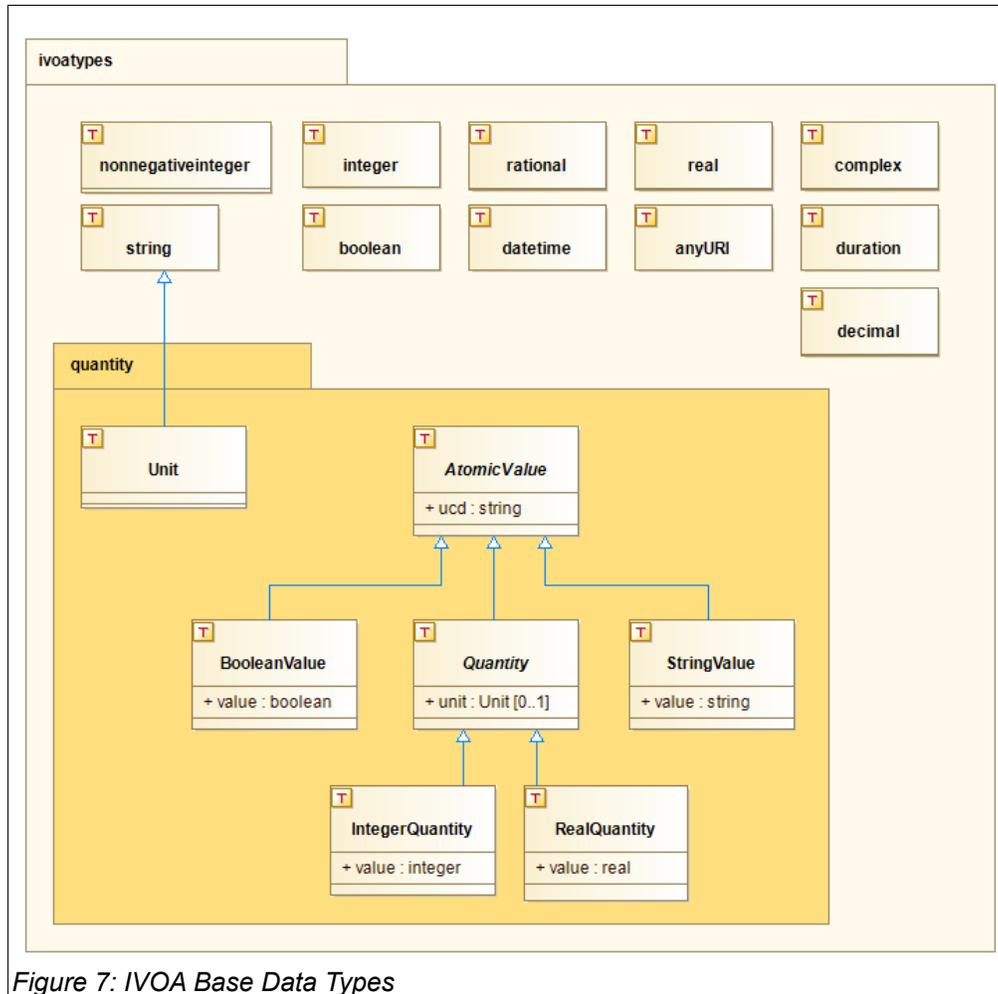


Figure 7: IVOA Base Data Types

6.1.1 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.

6.1.2 UCDs

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

6.1.3 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.

6.2 Cube Model DataTypes

This model defines no new data types or enumerations.

Appendix A: Modeling Conventions

This model adopts the same modeling conventions as outlined in Appendix A of the Dataset Metadata model. Here, we provide a brief summary of these statements.

1 Diagram notation

[TBD] STATEMENT.. same as Dataset Model?, reiterate here?

2 Model Identification metadata

We defined stereotypes on elements of this model to provide metadata identifying this model, and other models on which it is dependent.

2.1 Model stereotype

The Model stereotype (<<Model>>) consists of a set of Model properties which identify a particular model and its dependencies.

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.

3 Extensibility

User-defined content would be modeled as an extension of the IVOA standard.

Appendix B: N-Dimensional Cube Model Summary

Cube Model Identification				
Model Element	Datatype	Mult.	Meaning	value
Model identification				
Model				
Model.name	string	1	Data model name	"N-Dimensional cube"
Model.version	string	1	Data model version	"1.0"
Model.prefix	string	1	Data model prefix tag	"cube"
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	"STC"
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>

Cube Model Summary				
Model Element	Datatype	Mult.	Meaning	UCD1+
Cube Model Elements				
CoordSys			STC2 AstroCoordSystem	
DataProduct			Instance of data product (abstract)	
DataProduct.coordSys	CoordSys	*	Coordinate systems associated with data	
DataProduct.mappings	FrameMappings	0..1	Coordinate frame transforms	
FrameMappings			STC2 Mappings	
NDImageDataset			Instance of N-Dimensional image cube dataset	

Cube Model Summary

Model Element	Datatype	Mult.	Meaning	UCD1+
NDImageDataset.image	NDImage	1..*	N-Dimensional image arrays	
NDImage			N-Dimensional Image data	
NDImage.data	Voxel	*	N-Dimensional Image data array	
NDPoint			N-Dimensional data point	
NDPoint.spaceAxis	SpatialCoord	0..1	Data point value in spatial domain	
NDPoint.timeAxis	Time	0..1	Data point value in temporal domain	
NDPoint.spectralAxis	SpectralCoord	0..1	Data point value in spectral domain	
NDPoint.redshiftAxis	GenericRedshiftCoord	0..1	Data point redshift	
NDPoint.polarizataionAxis	PolarizationCoord	0..1	Data point polarization state	
PixelCoordSys			STC2 PixelCoordSystem	
PixelatedData			Instance of pixelated image data product (abstract)	
PixelatedData.pixelCoordSys	PixelCoordSys	1	Pixel coordinate system defintion.	
PointData			Instance of sparse data product (abstract)	
PointData.customAxes	GenericCoord	*	Generic coordinate point values	
SparseCubeDataset				
SparseCubeDataset.image	SparseCube	1..*	Instances of sparse cube data	
SparseCube			Instance of sparse cube data	
SparseCube.data	NDPoint	*	N-Dimensional sparse cube points	
Voxel			Volume pixel	
Voxel.pixelAxes	PixelCoordinate	1	Coordinate in nD pixel space	
Voxel.dependentAxis	Coordinate	0..1	Value at voxel coordinate	

References

- [1] "IVOA Dataset Metadata Model": Version 1.0, 10 Sept. 2014
<http://www.ivoa.net/Documents/...>
- [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>
- [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>