International
Virtual
Observatory
Alliance

The Observation Core Components Data Model Version 0.2

IVOA Working Draft January 15, 2010

This version:
http://www.ivoa.net/Documents/Notes/WD-ObsCoreDM-0.2-20100108.pdf

Latest version:
http://www.ivoa.net/Documents/latest/ObsCoreComponentsDM.html

Previous versions:
0.1

Editor(s):
Mireille Louys, François Bonnarel

Authors:
Mireille Louys, François Bonnarel, Alberto Micol,
David Schade, Anita Richards, Pat Dowler
Jesus Salgado, Doug Tody, Gerard Lemson,
Igor Chilingarian, Daniel Durand, Randy Thomson, Gretchen Green

Abstract

This document discusses the definition of the core components of the Observation data model that are necessary to cover data discovery use-cases when querying data centers. It exposes the use-cases to be carried out, explains the model and provides a table of fields to be implemented along the lines of a TAP/SCHEMA strategy. Such a small model is easy to understand and implement by data providers that wish to publish their data into the Virtual

Observatory.

Status of this document

This document has been produced by the Data Model Working Group. It is still a draft to be improved within the dm working group.

Acknowledgements

ML and FB are grateful for Euro-VO AIDA fundings

Contents

1 Introduction

Modeling of observational metadata has been a long term activity in the IVOA since it was created in 2002. Various modeling efforts like Resource Metadata, STC, Spectrum data model, and Characterisation data model, have been recommended and are currently used in IVOA services and applications. Historically, models and protocols have been developed in parallel and first focused on simple data types and simple protocols accordingly. However the guide line in the DM WG was to foster full interoperability by covering the full chain of actions a user might want to do for his/her science: data-discovery, data retrieval, data analysis. This work comes now to a more mature state where we need to homogeneise the various approaches in oder to discover/retrieve/analyse all kinds of observation data products.

Although there has been great successes in the use of some of the data models (Resource metadata, Spectrum with SSA) the general approach described above has this drawback that it ends up as relatively large data models that people fill difficult to implement and use. This is peculiarly the case of the ObsDM and inside it of one of its packages the Characterisation Data model, which is an IVOA recommendation [3] (Louys et al, 2007). This situation is also reinforced by more technical problems: serialisation (pure XML or Utypes), and protocols for metadata access were not always available for practical implementation of these data models. A contrario that's probably also why Resource metadata and Spectrum have been implemented by data centers. Their strong linkage with Registry and SSA protocols explains somewhat their relative success. In case of Observation and Characterisation data model one obvious family of use cases has long been Data discovery. Up to recently, however we were lacking a protocol to convey metadata organized along the lines of this model from data archives and services to client software and services. SIAV2 has long been a candidate to permit this and still is in the case where archives are containing images and cubes, but the emergence of the TAP protocol allows now to use a generic method for all kind of datasets and models. Definition of TAP services implementing the Core of Observation data model for various archives will provide a unified discovery interface to a large set of heterogeneous data archives of images, cubes, spectra and catalogs. Although this effort is called hereafter "ObsTAP definition" for obvious reasons, this approach is consistent with the GDS protocol concept as described in the DAL architecture document ([1] . The goal of the current note is to propose a subset of Utypes of the overall Observation data model currently under construction sufficient to fulfill large categories of use-cases collected from the community. This was an incentive launched by David Shade under the auspices of the IVOA Take-up committee , in May 2009.

The document is composed as following: Section 2 will present these use cases. Section 3 will summarize briefly the Observation/Characterization data model effort and identify useful features for our purpose. Section 4 defines 2 tables of data model fields, with their name, Utype in the Observation Core components DM, Units, type and a short description to be used in a TAP/Schema implementation for data discovery queries matching the Use-cases. Section 5 gives a general outline for binding an abstract logical representation of the necessary metadata, the Observation Core components model to its RDBMS representation using TAP/ADQL protocol.

Appendix A contains the Use-cases gathered from the community and listing various possible search criteria in the data discovery process. Appendix B provides examples of ObsTAP queries and responses with different levels of complexity. Appendix C illustrates other possible serialisation of the data model as for instance XML document examples.

2 Use cases

We first focused on data discovery use-cases, aimed at finding observations in the VO domain, by broadcasting the same query to a bunch of data centers or to all VO subscribers.

Ultimately we need to provide data providers with a list of item and features that they could easily map to their database system, in order to answer to the defined queries.

The goal is to be simple enough to be implementable, and not to be exhaustive on all exotic data sets.

The main features of these use-cases are mainly: - multi-wavelength search - multi-types of data (spectrum, cube , catalogs) Refined or advanced searches may include extra knowledge stemming from astronomical objects classification and would need to extract results from catalogs, possibly by using fine sub-queries.

Here we list just one example in each use-case category but the full list is available in Appendix I or at http://....

2.1 Discover imaging data of interest

Use-Case 1.2: Let me input a list of RA and DEC coordinates and show me spatially coincident data that satisfies

1.2.1 Data type is Imaging or spectroscopy data
1.2.2 Includes one or more of the RA,DEC

1.2.3 Includes both a wavelength in the range 5000-900 angstroms AND an X-ray image (AND=SERVREQ)

2.2 Discover Spectral data of interest

Use-case 2.2: Show me a list of all data that satisfies

2.2.1 DataType=Spectrum
2.2.2 Wavelength includes 6500 angstroms
2.2.3 Spectral Resolution better than 15 angstroms
2.2.4 Spatial Resolution better than 2 arcseconds FWHM
2.2.5 Exposure Time ¿ 3600 seconds
2.2.6 Data Quality = Any

2.3 Discover Data cubes of interest

Use-Case 3.4 : Show me a list of all data that satisfies

3.4.1 DataType=cube
3.4.2 RA includes 16.00
3.4.3 Dec includes +41.00
3.4.4 Wavelength includes 6500 angstroms
3.4.5 Wavelength includes 4000 angstroms
3.4.6 Spectral resolution better than 5 angstroms
3.4.7 Exposure time more than 3600 seconds
3.4.8 Data Quality= Fully Calibrated

2.4 Discover general data of interest

Use-Case 5.3 : Show me a list of all data that satisfies

5.3.1 DataType=Imaging or Spectroscopy
5.3.2 RA includes 16.00 hours
5.3.3 DEC includes +41.00 degrees
5.3.4 SDSS images and spectra AND CFHTLS images and spectra

A common denominator to most of the Use-cases defined here is the set physical features in terms of spatial, spectral, temporal, and photometric properties? This is well covered by the Characterisation Data Model and can be re-used as a package within the Observation DM. The Characterisation DM as well as the Spectrum DM define lists of Utypes [4] that are ready to use for our cases. Access and identification metadata, defined and summarized in the SSA Utype list [5], provide the necessary mecanism to identify and retrieve a data set.

2.5 How to answer these use cases?

The idea is to build up a VO interface to allow data providers to simply describe their observation metadata stored in DBMS systems and provide a query mechanism for users to discover , then retrieve the VO enabled data sets. The currently developed TAP protocol offers the appropriate implementation layer. opportunity to do that, because it provides a generic method to implement an IVOA data model in a database by the usage of utypes describing the model classes and attributes and linked to the columns. This can be described in a standard way by the TAP/Schema. Queries to this database may be done in all cases by ADQL and later in some cases via PQL. The way Utypes are generated from a data model is not described here. We assume it will be consistent with the Utype definition WD (Louys et al, 2009).

3 The Observation Core Components Data Model

Let us just sketch out how the core components fit into the scope of the general Observation Data Model and list which classes and attributes will be used to support the above use-cases.

3.1 The context and history of metadata modeling in the VO

The Observation data model project appeared at the first Data Model forum held at the May 2003 IVOA meeting in Cambridge,UK. Rapidly some main classes appeared to be necessary to organize the metadata: Dataset or Observation, Identification, Physical Characterisation, Provenance (either instrumental or software) and Curation. A description of the early stages of this development can be found in [2],(Observation IVOA note). In parallel an effort dedicated to spectra was lead by the DM Working group. The Spectrum data model represents all necessary metadata for one specific type of observational data: simple spectra. For the overall Observation Data Model, the physical characterization has been identified to be on first priority already in 2004. It was completed as an IVOA recommendation after 4 years of discussion which included computer scientists, astronomers and data providers under the lead of J.Mc Dowell.

3.1.1 Characterisation data model

The Characterisation data model organizes metadata as a 3D matrix spanning independently the various physical axes (spatial, spectral, time, flux or whatever observable quantity), four levels of granularity, and some features or Properties (coverage, resolution and sampling). This scheme allows to support selection of data sets for data discovery as well as data analysis.

3.1.2 New efforts

While the Characterisation data model was setting up a logical framework to describe the properties/features of each observation in the VO, the approach here is more pragmatic and leads to a simple implementation data model, and its protocol application using the emerging TAP/SCHEMA framework. In the mean time, the Generic Data set/Observation data model is currently developed integrating Curation and DataID description borrowed from the Spectrum Data model with a detailed description of the Provenance (instrumental and computational) for observed datasets. The consistency between the two efforts is a major goal, and will be warranted by defining the Core components to be re-usable at the more general level.

3.2 Observation Core Components data model and attributes definition

From all the metadata covered and described in the full Observation data model, only a small part is needed to support data discovery in a regular and efficient manner. We then concentrate first on the definition of core components of the Observation data model that will be used to support the use-cases described above. The Observation Core component data model is summarized in a UML class diagram in next section and described as a logical data model. The implementation of such a model using the TAP/SCHEMA framework(/binding) will make use of one of the two tables : `ivoa.ObsCore` and `ivoa.ObsCore_ext` in order to represent and exchange metadata necessary to satisfy the data discovery use-cases defined above.

Here we list all the metadata used as selection criteria in the listed use-cases and in the response data center might give back.

3.2.1 UML description of the model

The data model for observation is organised according to some Object Oriented Programming principles in order to define unique and consistent concepts , as re-usable classes. UML helps to sketch out the class organisation as shown in Fig. 1. This class diagram covers all classes used in the context of the Observation Core components Model. The Characterisation classes, describing how the data span along the main physical axes, are used here partly, and only the attributes relevant to this modeling effort are shown here. This is also the case for the Dataid and Curation classes extracted from the Spectrum/SSA data model where only a subset of attributes are necessary for data discovery.We consider for now that we use Characterisation classes only down to the Support level.

Encoding the coordinates attributes depends on the nature of each Characterisation axis and will be described in detail in the Full Observation DM. The Utypes shown on both tables provide the inner structure of each class attribute. Therefore we do not develop the SpatialAxis, TimeAxis classes on the diagram , for the sake of clarity.

4 Data Model detailed description

Here we try to describe the data model attributes and classes and mention the corresponding "short name" that would be used in the OBS/TAP implementation of this model.

4.1 Observation

This class is a place holder that gathers all metadata relative to an observed and distributed data set. It points to existing classes of Spectrum DM and VODataservice.

4.1.1 Observation data set type

The model defines a *data product type* attribute for the Observation Class. It is the type of observation the user queries for or selects for retrieval. This is coded as a string that conveys a general idea of the content and organisation of a data set. We consider a coarse classification of the types of data set interesting for science usage, covering images, cubes, spectra, light curves, SED, etc... The Observation.ProductType attribute takes its string value in the following set, organized according to up to 4 levels of granularity:

- **Image**
  - Image.2DSkyImage
  - Image.2D any 2D image: weight maps, bad-pixel map, etc: TBdiscussed
  - Image.Longslit 2D image for a long slit spectrum with one axis mapped on wavelength
  - Image.Cube 2D+ extra dimension(s)
  - Image.Cube.Spectral
  - Image.Cube.Time

- Image.Cube.Polarization

- **Spectrum**
  - Spectrum.1D
  - Spectrum.SED
  - Spectrum.Polarisation (TO BE DISCUSSED)...
  - Spectrum.Echelle
  - Spectrum.IFU (to be discussed)
- **TimeSeries**
  - TimeSeries
  - TimeSeries.LightCurve flux variable with Time
  - TimeSeries.RadialVelocity ???(to be discussed)
- **Visibility**
  - Visibility.Image
  - Visibility.Cube
- **EventList**

This provides a hierarchy of possible data product types that is stated here but can be extended by data providers in the future depending to new kinds of data and search procedures.. The first string before the first '.' can be used as a *primary data product type* and should be supported by all data centers. The short name for this attribute is `data_product_type`) in the protocol item list, as shown in Table 3.

4.1.2 Observation Calibration level

It is a convention we suggest to use to classify the different possible calibration status of an observed data set. This is up to the data provider to consider how to map his own internal classification to the suggested scale here.

Following examples can help to find the most appropriate value for the *calibLevel* attribute.

- Level 0:
  raw instrumental data, possibly in proprietary internal provider format, that need specific tools to be handled.
- Level 1:
  In a standard format(FITS, VOTable, etc...) but not fully calibrated. Standards tools can handle it.
- Level 2:
  Science ready data , with instrument signature removed, and physical units on all physical axes, for instance spatial in deg , flux in Jy, spectral in KeV

- Level 3:
  Enhanced data products like mosaics, improved co-added image cubes, resampled or drizzled images, etc. spectra with calibrated velocity axis at a particular line rest frequency. In such case, the improved calibration procedure is described by the data provider in some way, progenitors of such a data product can be identified into the reduction pipeline.

This classification is simple enough to cover all regimes. Data providers will adjust the mapping of their various internal levels of calibration to this general frame, with the knowledge of the PIs for each project.

4.1.3 Observation regime

The spectral regime to which an observation belongs is modeled as *obs:Observation.waveband* with short name : `em_domain` . This re-uses the waveband definition from the VODataService IVOA standard and the same set of enumerated strings: RADIO, MILLIMETER, INFRARED, OPTICAL, UV, EUV, X-RAY, GAMMA-RAY available at
http://www.ivoa.net/Documents/VODataService/20090903/PR-VODataService-1.1-20090903.pdf

4.1.4 Identification

After acquisition and reduction an observation is uniquely identified by its creator and gets a creator data set identifier. This information is defined in the Spectrum data model in the DataID class. We re-use the 'DataID.CreatorDID' string in order to distinguish two data set curated by two different services (archives) but originating from the same creator. When broadcasting a query to multiple servers, the response may contain multiple copies of of the same data set, therefore a unique identifier is needed here. The second identifier used in this model is the one given by the data provider, 'Curation.PublisherDID' as defined in the Resource Metadata and should also be provided by the service.

4.1.5 Target

This is the astronomical object of interest for which the observation was performed. Only the name is used in the Core Components model but the Target object is fully designed in the Spectrum data model. Serendipitous archives or surveys may not contain this information for all observations, so this can be set to "not defined". The target may be a special type of observation like "dark" or '' CCD"??

4.1.6 Description of physical axes: Characterisation classes

As mentioned in the use-cases, selection criteria for an observation depend on the physical axes contained in the data set especially the position, band , time, and the type of observed quantity, that we call "observable" in the data model and can represent various types of flux but also velocity, etc. Such a description was tackled in the IVOA Characterisation data model from which we re-use mainly the 2 first level of details except for the spatial coverage where the support region (level 3) is used.

1. Spatial axis

   - The observation reference position
     Two coordinates in position are used to identify a reference position (typically the center) of an observation in the sky, attached to a coordinate system definition. The coordinate system is defined in the Characterisation DM, based on STC:Coordsys. In this implementation of the model we state that coordinates are given in ICRS.
     `s_ra` is used as shortname for the fully expanded Utype in the Characterisation DM:
     obs:Char/SpatialAxis.Coverage.Location.coord.Position2D.Value2.C1.
     `s_dec` maps the Utype :
     obs:Char/SpatialAxis.Coverage.Location.coord.Position2D.Value2.C2.
   - The covered region
     It is modeled either as a couple of intervals on each coordinates : `s_ra_min` obs:Char/SpatialAxis.Coverage.Bounds.limit
     `s_ra_max` obs:Char/SpatialAxis.Coverage.Bounds.limits.CoordInterval.HiLimit2Vec.C1
     `s_dec_min` obs:Char/SpatialAxis.Coverage.Bounds.limits.CoordInterval.LoLimit2Vec.C2 `s_dec_max`
     obs:Char/SpatialAxis.Coverage.Bounds.limits.CoordInterval.HiLimit2Vec.C2
     or as a full region that defines more precisely where the spatial footprint of the data set lies on the sky.
     The shortname is `s_region` and mapped to
     obs:Char/SpatialAxis.Coverage.Support.Area stored as an STC region.
   - The spatial resolution
     The minimal size that can be distinguished along the spatial axis, `s_resol` is stored in arcseconds and is called
     obs:Char/SpatialAxis.Resolution.RefVal in the observation Utype list.

2. Time axis
   Three time stamps are used: `t_start` , `t_stop` , and `t_span` the elapsed time. A format like MJD is useful for easy calculations and preferred for the observation Core components models. The elapsed time is expressed in days.
   `t_resolution` provides the FWHM value of the temporal resolution .
3. Spectral axis
   This axis is generally used to represent different kinds of physical measurements: wavelength, energy, frequency. or some interpretation of this with respect to a reference position like velocity. The data model distinguishes the various flavors of this axis using the ucd attached to it, *obs:Char/SpectralAxis.ucd* or `em_ucd` .
   `em_min` and `em_max` gives the bounds of data present along the spectral axis. Such values are expressed as frequencies but using meters as units as it is easily convertible. The resolution power along this axis is called `em_res_power` and maps to the
   obs:Char/SpectralAxis.Resolution.ResolPower Utype in Characterisation DM.
4. Observable axis
   What was observed on this axis is described by a UCD string which can be any of :phot.flux, phot.flux.density, phot.count, phot.mag, or more complex combinations like phot.flux.density;phys.polarization.stokes.I .

4.2 Provenance

4.2.1 Instrument name

`instrument` provides the name of the instrument used for the acquisition of the observation. It is given in the model as *obs:Provenance/ObsConfig.instrument.name*and encoded as a string. More observation details on the instrumental configuration are available in the extension list, below 4.

4.2.2 Access to the data

The data format as well as the url to access the data files are provided by the *Access* Class inherited from the SSA Utype list and mapped to `mime_type` and `access_url` keywords respectively.

5 Identification of metadata and their Utypes for Generic Data Discovery

Figure 3 presents the table of mandatory fields which must be supported by an Obs/TAP service. It is a list of items with a suggested short name , to be used for the TAP tables column names, the corresponding Utype in the Observation Core Components model, a unit name in which that field will be expressed, the type of the field, and a short description of this metadata .

These metadata items make up the minimal set of fields to be handled by a service supporting the use-cases exposed in section 2 and compliant to the data model.

More detailed queries, especially on the observable axis, etc. can be supported as well. They would use complementary items listed in the optional table shown at Fig.4.

Most observation measure some flux quantity depending on position and/or wavelength/energy/frequency and or time. Here we consider a more general axis : the "Observable axis" that can be either flux or any other quantity, the nature of which is given by the UCD attached to this axis.

For instance :

observable	ucd	units
flux	phot.flux	Jy
radial velocity	phys.veloc	km/s
Stockes U	phot.flux;phys.polarisation	Jy

6 Hints for an implementation in TAP

Minimal requirement to build up an Obs/Tap service will be to define which fields to query on. We consider two tables in a single schema: `ivoa.obscore` and `ivoa.obscore_ext`. `ivoa.obscore` lists the minimum metadata set of queriable parameters necessary to support the most common use-cases. `ivoa.obscore_ext` includes the first table and adds more parameters prone to be used in more advanced queries.

The schema describes the columns (as in table 1 and 2 given above) and will allow clients to make appropriate linkage between column names, units, UCDs and observation data model Utypes. See TAP recommendation for more details on what that does mean. In practice these tables can be implemented either as real tables or as VIEWS built on top of an Observation index or Observation log or any kind of service providing metadata for Observations. A query response table can be reused for further queries to another Obs/TAP service. Joint statements will allow to add more columns to the response from the data base proprietary data model. The observation identifier of the Obs/TAP query response will be the key query parameter to query other tables in the service, while the `access_url` field will allow direct retrieval of the dataset. This URL is simply provided within the "ObsCore" table coming back as query response. The data retrieval itself will be up to the application interpreting the response.

References

[1] D.Tody et al. Ivoa data access layer service architecture and standard profile. *http://www.ivoa.net/internal/IVOA/SiaInterface/DAL2_Architecture.pdf*, 2009.

[2] J. McDowell et al. Observation data model for astronomical dataset. *http://www.ivoa.net/Documents/Notes/????????????????.pdf*, 2005.

[3] M. Louys et al. Data model for astronomical dataset characterisation. *http://www.ivoa.net/Documents/latest/CharacterisationDM.html*, 2007.

[4] Data model WG. Utype list for the characterisation data model v1.11. *http://www.ivoa.net/Documents/Notes/UTypeListCharacterisationDM/UtypeListCharacterisationDM-20070522.pdf*, 2007.

[5] DAL WG. Simple spectral access protocol. *http://www.ivoa.net/Documents/latest/SSA*, 2007.

Appendix A: Data discovery Use-cases

Use-cases to be included here see original document.

Appendix B: Example of ADQL queries for some of our use cases.

Here we consider a very general use case from the list (1.1) :
Show me a list of all data that satisfies:

1.1.1 Datatype=any
1.1.2 contains RA=16.0 and DEC=40.0

These data would be searched on all VO services by sending the following query:

```
SELECT * FROM ivoa.Obscore WHERE
s_ra_min < 16.0 AND s_ra_max > 16.0 AND
s_dec_min < 40.0 AND s_dec_max > 40.0
```

Here is the query response example that could be delivered by a service supporting Obs/TAP.

More constraints can be added in the following use-case (1.3):
Show me a list of all data that satisfies

1.3.1 DataType=Image
1.3.2 Spatial resolution better than 0.3 arcseconds
1.3.3 Filter = J or H or K
1.3.4 RA between 16 hours and 17 hours
1.3.5 DEC between 10 degrees and 11 degrees

```
SELECT * FROM ivoa.Obscore_ext WHERE
dataproduct_type=‘‘Image.2D’’
AND  s_resolution < 0.3
AND s_ra > 240 AND s_ra < 255 AND
s_dec > 10 AND s_dec < 11
AND
(em_min > 2.1 AND em_max < 2.4) OR
(em_min >= 1.6 AND em_max <= 1.8) OR
(em_min >= 1.2 AND em_max <= 1.4)
```

Here (Fig. 6 is the query response delivered by a service supporting Obs/TAP.

Appendix C: XML serialisation of the same Use-case

...here we would add an XML tree equivalent to the query response for the same use-case (1.3) just to illustrate that some other serialisation can be generated as well.

Appendix D: Updates of the document

- version 0.1 to 0.2
  - plenty of changes in the tables/ utypes stabilisation
  -
- version 0.2 to 0.3
  -
  -

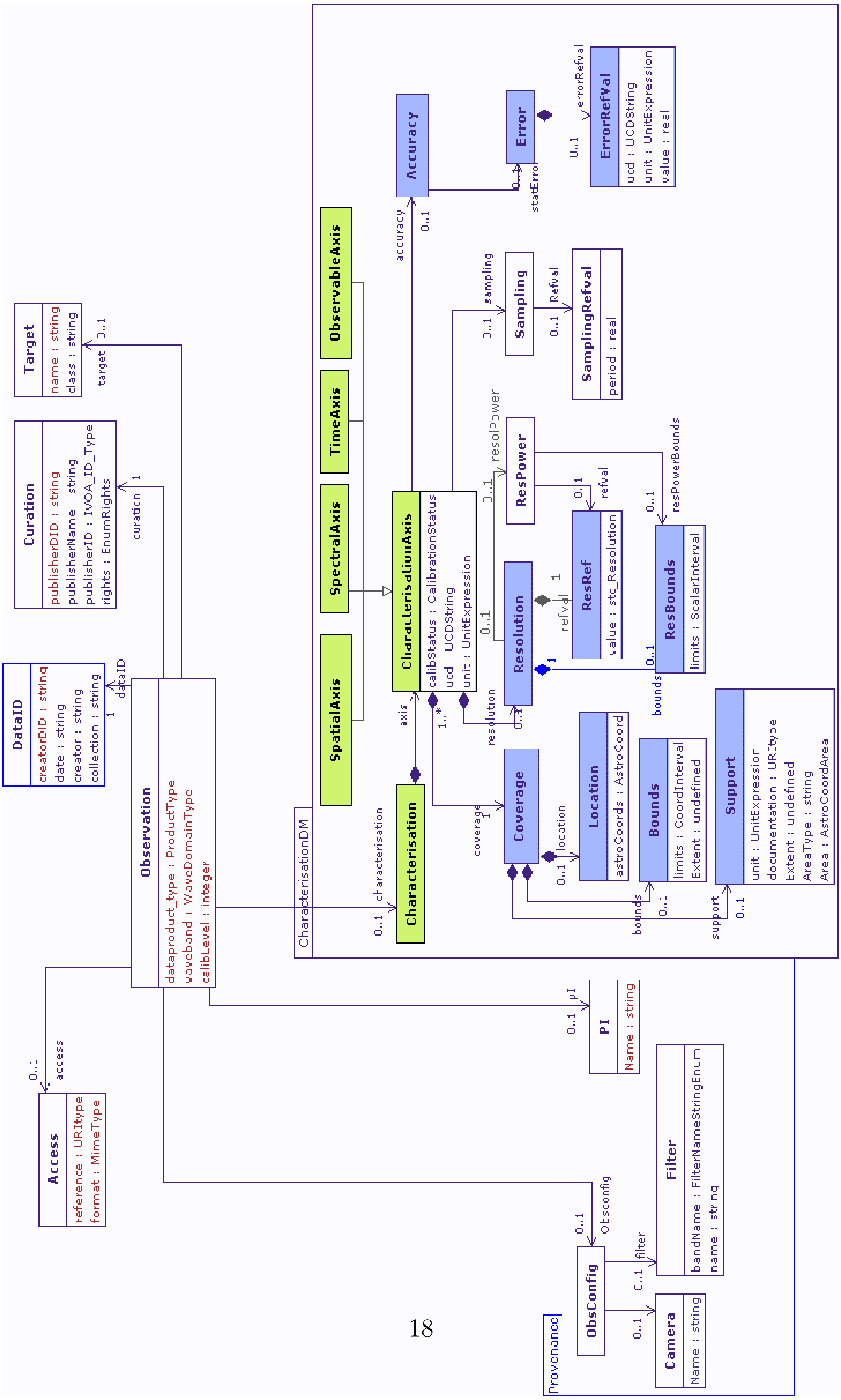

Figure 1: *Here is the class diagram representing the classes used to organise observational metadata. Classes may be linked together via an association or aggregation link. The minimal set of necessary attributes for data discovery is shown in brown.*

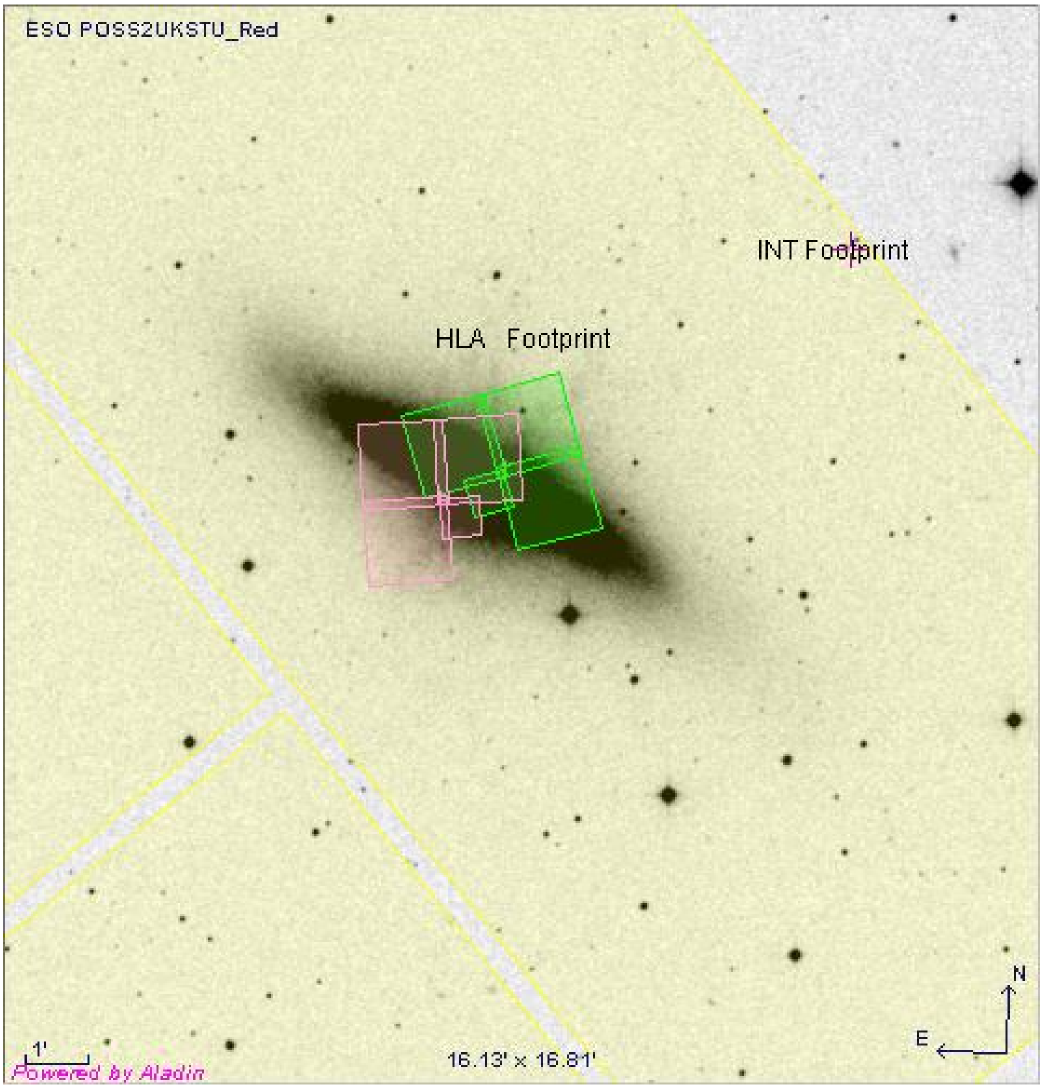

Figure 2: *This figure illustrates which information is relevant to the spatial coverage of an observation. A coarse region description is given as 2 [min, max] coordinates intervals , like the bounding box parallel to the axes, while the real shape of the instrument footprint is given as the support, in a region description based on the STC region class.*

Local short name	Utype	Units	Type	Description
dataproduct_type	obs:Observation.ProductType	unitless	enum	see proposal
obs_publisher_did	obs:Curation.PublisherDID	unitless	string	Data set ID given by the publisher.
obs_creator_did	obs:DataID.CreatorDID	unitless	string	Ivoa ID given by the creator
calib_level	obs:Observation.calibLevel	unitless	enum integer	Calibration status of the observation: in {0, 1, 2, 3}
target_name	obs:Target.name	unitless	string	object of interest
s_ra	obs:Char/SpatialAxis.Coverage.Location.coord.Position2D.Value2.C1	[deg]	double	Central Spatial Position in ICRS
s_dec	obs:Char/SpatialAxis.Coverage.Location.coord.Position2D.Value2.C2	[deg]	double	
s_ra _min	obs:Char/SpatialAxis.Coverage.Bounds.limits.CoordInterval.LoLimit2Vec.C1	[deg]	double	Min ra coordinates of spatial bounding box in ICRS
s_ra _max	obs:Char/SpatialAxis.Coverage.Bounds.limits.CoordInterval.HiLimit2Vec.C1	[deg]	double	Max RA coordinates of spatial bounding box in ICRS
s_dec_min	obs:Char/SpatialAxis.Coverage.Bounds.limits.CoordInterval.LoLimit2Vec.C2	[deg]	double	Min DEC limit in spatial Position in ICRS
s_dec_max	obs:Char/SpatialAxis.Coverage.Location.limits.CoordInterval.HiLimit2Vec.C2	[deg]	double	Max DEC limit in spatial Position in ICRS
s_resolution	obs:Char/SpatialAxis.Resolution.refVal	[arcsec]	float	Spatial resolution of data
t_start	obs:Char/TimeAxis.Coverage.Bounds.Limits.TimeInterval.StartTime	MJD	double	Start time
t_stop	obs:Char/TimeAxis.Coverage.Bounds.Limits.TimeInterval.StopTime	MJD	double	Stop time
t_span	obs:Char/TimeAxis.Coverage.Bounds.Extent	day	float	Total observation elapsed time
t_exptime	obs:Char/TimeAxis.Coverage.Support.Extent	[s]	float	Total exposure time
t_resolution	obs:Char/TimeAxis.Resolution.refVal	[s]	float	Temporal resolution FWHM
em_domain	obs:Observation.waveband	unitless	enum string	vr:waveband in { RADIO, MILLIMETER, INFRARED, OPTICAL, UV, EUV, X-RAY, GAMMA-RAY } cf VODataservice
em_min	obs:Char/SpectralAxis.Coverage.Bounds.limits.Interval.LoLim	[m]	double	start in spectral coordinates
em_max	obs:Char/SpectralAxis.Coverage.Bounds.limits.Interval.HiLim	[m]	double	stop in spectral coordinates
em_res_power	obs:Char/SpectralAxis.Resolution.ResolPower	unitless	double	Value of the resolution power along the SpectralAxis.
o_ucd	obs:Char/ObservableAxis.ucd	unitless	string	Nature of the observable axis; necessary for polarision data or any kind of flux. Values in { phot.flux, phot.flux.density, phot.count, phot.mag }
instrument	obs:Provenance/ObsConfig..instrument.name	unitless	string	In uppercase ?
access_url	obs:Acces.Reference	unitless	uri string	URL used to access dataset

Figure 3: *Table of mandatory data model items necessary to build-up TAP queries for the targeted data discovery use-cases.*

ShortName	Utype in the Observation CoreComponents DM	Unit	Type	Description
DATAID				
creation_date	obs:DataID.Date	unitless	date	Format ISO8601 or MJD ?
obs_collection_name	obs:DataID.Collection	unitless	string	Or archive name
obs_creator_name	obs:DataID.Creator	unitless	string	Name of the creator of the data
TARGET				
target_class	obs:Target.Class	unitless	string	Class of the Target object as in SSA
CURATION				
publisher_name	obs:Curation.Publisher	unitless	string	Archive service ,CADC,CDS,StSci
publisher_id	obs:Curation.PublisherID	unitless	string	ivoaID for the Publisher
bib_reference	obs:Curation.Reference	unitless	string	Service bibliographic reference
data_rights	obs:Curation.Rights	unitless	enum	Public/Reserved/Proprietary/
nb_members	obs:Observation.numsegm	unitless	integer	Nb of obs. elements in an association
CHARACTERISATION				
space				
s_ucd	obs:Char/SpatialAxis.ucd	unitless	ucd string	(pos or u,v data)
s_region (s_footprint)	obs:Char/SpatialAxis.Coverage.Support.Area		stc:AstroCoord Area	Region covered in STC or ADQL
s_resolution_bound_min	obs:Char/Spatial.Resolution.bounds. Limits.Interval.LoLim	arcsec	double	Resolution min value on spectral axis (FHWM of PSF)
s_resolution_bound_max	obs:Char/Spatial.Resolution.bounds. Limits.Interval.LoLim	arcsec	double	Resolution max value on spectral axis
astrometric_cal_status	obs:Char/SpatialAxis.calibStatus	unitless	enum	NOT CALIBRATED, FINE, COARSE
astrom_precision_stat	obs:Char/SpatialAxis.Accuracy.StatError.errorRefVal.value	arcsec	double	Astrometric precision along the spatial axis
s_pixel_scale	obs:Char/SpatialAxis.Sampling.refVal.period	arcsec	double	Pixel spacing in spatial units
time				
t_cal_status	obs:Char/TimeAxis.calibStatus	unitless	enum	Type of coord calibration
t_staterr	obs:Char/TimeAxis.Accuracy.StatError.errorRefVal.value	s	double	Time coord statistical error
spectral				
em_resol	obs:Char/SpectralAxis.Resolution.refVal.value	m	double	Value of Resolution along the SpectralAxis
em_resolPower_min	obs:Char/Spectral.Resolution.bounds. Limits.Interval.LoLim	m	double	Resolution power min value on spectral axis
em_resolPower_max	obs:Char/Spectral.Resolution.bounds. Limits.Interval.LoLim	m	double	Resolution power max value on spectral axis
em_stat_err	obs:Char/SpectralAxis.Accuracy.StatError.errorRefVal.value	m	double	Spectral coord statistical error
observable				
o_cal_status	obs:Char/ObservableAxis.calibStatus	unitless	enum	Level of calibration for the observable coord
o_ucd	obs:Char/ObservableAxis.ucd		UCD string	UCD for observable;
o_detection_limit	obs:Char/ObservableAxis.Resolution.refval	?	double	Average resolution along observable
o_stat_err	obs:Char/ObservableAxis.Accuracy.StatError.errorRefVal.value		double	Observable statistical error
PROVENANCE				
PI_name	obs:Provenance/PI.name	unitless	string	Name of Principal Investigator
filter_band	obs:Provenance/ObsConfig.Filter.bandName	unitless	string	For instance : U, B u, g i, k
filter_name	obs:Provenance/ObsConfig.Filter.name	unitless	string	Filter name as stated into the archive: e.g FW66
camera_name	obs:Provenance/ObsConfig.camera.name	unitless	string	Name of camera
optical_element_name	obs:Provenace/ObsConfig.opticalElem.name	unitless	string	Name of optical element
telescope_name	obs:Provenance/ObsConfig.telescope.name	unitless	string	Name of telescope

Figure 4: *Table of optional complementary fields to be used for OBS/TAP queries.*

```
<?xml version="1.0" encoding="UTF-8"?>
<VOTABLE xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="xmlns:http://www.ivoa.net/xml/VOTable/VOTable
-1.1.xsd"
 xmlns:ssa="http:www.ivoa.net/xml/DalSsap/v1.0" version="1.1">
<RESOURCE type="Results">
<DESCRIPTION>ObsCoreDM/GDS TAP service</DESCRIPTION>
<INFO name="QUERY_STATUS" value="OK"/>
<INFO query="obs:Char/SpatialAxis.coverage.limits contains (16.0,+41.0)" />
<TABLE>

<FIELD ID="conty" name="dataproduct_type" datatype="char" arraysize="*"
utype="Observation.ProductType">
</FIELD>
<FIELD ID="calev" name="calib_level" datatype="char" arraysize="*"
utpe="obs:Observation.calibLevel">
</FIELD>
<FIELD ID="waveb" name="em_domain" datatype="char" arraysize="*"
utype="obs:Observation.waveband">
</FIELD>
<FIELD ID="pubdid" name="obs_publisher_did" datatype="char" arraysize="*"
utype="obs:Curation.PublisherDID">
</FIELD>
<FIELD ID="crdid" name="obs_creator_did" datatype="char" arraysize="*"
utype="obs:DataID.CreatorDID">
</FIELD>
<FIELD ID="acref" name="access_url" datatype="char" arraysize="*"
utype="obs:Access.Reference">
</FIELD>
<FIELD ID="targe" name="target_name" datatype="char" arraysize="*"
utype="obs:Target.name">
<DESCRIPTION>Target pointed for this data set or misc ( Dark, etc...)
</DESCRIPTION>
</FIELD>
<FIELD ID="spara" name="s_ra" datatype="double" ucd="pos.eq.ra"
utype="obs:Char/SpatialAxis.Coverage.Location.coord.Position2D.Value2.C1"
unit="deg">
<DESCRIPTION>Spatial Position RA</DESCRIPTION>
</FIELD>
<FIELD ID="spdec" name="s_dec" datatype="double" ucd="pos.eq.dec"
utype="obs:Char/SpatialAxis.Coverage.Location.coord.Position2D.Value2.C2"
unit="deg">
<DESCRIPTION>Spatial Position DEC</DESCRIPTION>
</FIELD>
<FIELD ID="stara" name="s_ra_min" datatype="double" ucd="pos.eq.ra"
utype="obs:Char/SpatialAxis.Coverage.Bounds.limits.CoordInterval.LoLimit2Vec.
C1" unit="deg">
<DESCRIPTION>Bounds Position RA</DESCRIPTION>
</FIELD>
<FIELD ID="stadec" name="s_dec_min" datatype="double" ucd="pos.eq.dec"
utype="obs:Char/SpatialAxis.Coverage.Bounds.limits.CoordInterval.LoLimit2Vec.
C2" unit="deg">
<DESCRIPTION>Bounds Position RA</DESCRIPTION>
</FIELD>
<FIELD ID="stora" name="s_ra_max" datatype="double" ucd="pos.eq.ra"
utype="obs:Char/SpatialAxis.Coverage.Bounds.limits.CoordInterval.LoLimit2Vec.
C1" unit="deg">
<DESCRIPTION>Bounds Position RA</DESCRIPTION>
```

Figure 5: *Identifying pieces of a data model: Example of a possible Query response returned for the first query example, as a VOTable document.*

```xml
<?xml version="1.0" encoding="UTF-8"?>
<VOTABLE xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="xmlns:http://www.ivoa.net/xml/VOTable/VOTable
-1.1.xsd">
<RESOURCE type="Results">
<DESCRIPTION>ObsCoreDM/GDS TAP service</DESCRIPTION>
<INFO name="QUERY_STATUS" value="OK"/>
<INFO query="obs:char.SpatialAxis.coverage.limits contains (16.0,+41.0)" />

<TABLE>

<FIELD ID="publi" name="publisher_name" datatype="char" ucd="meta.curation"
utype="obs:Curation.Publisher" arraysize="*">
<DESCRIPTION>Dataset publisher (string)</DESCRIPTION>
</FIELD>
<FIELD ID="pubid" name="publisher_id" datatype="char" ucd="meta.curation"
utype="obs:Curation.PublisherID" arraysize="*">
<DESCRIPTION>Dataset publisher ivoa ID</DESCRIPTION>
</FIELD>
<FIELD ID="refer" name="bib_reference" datatype="char"
ucd="meta.bib.bibcode" utype="obs:Curation.Reference" arraysize="*">
<DESCRIPTION>URL or Bibcode for documentation</DESCRIPTION>
</FIELD>
<FIELD ID="right" name="data_rights" datatype="char"
utype="obs:Curation.Rights" arraysize="*">
<DESCRIPTION>Restrictions on data access</DESCRIPTION>
</FIELD>
<FIELD ID="pudid" name="obs_publisher_did" datatype="char"
ucd="meta.ref.url;meta.curation" utype="obs:Curation.PublisherDID"
arraysize="*">
<DESCRIPTION>Publisher's ID for the dataset </DESCRIPTION>
</FIELD>

<FIELD ID="creat" name="obs_creator_name" datatype="char"
utype="obs:DataID.Creator" arraysize="*">
<DESCRIPTION>Dataset creator</DESCRIPTION>
</FIELD>
<FIELD ID="colle" name="obs_collection_name" datatype="char"
utype="obs:DataID.Collection" arraysize="*">
<DESCRIPTION>Data collection to which dataset belongs</DESCRIPTION>
</FIELD>
<FIELD ID="crdid" name="obs_creator_did" datatype="char" ucd="meta.id"
utype="obs:DataID.CreatorDID" arraysize="*">
<DESCRIPTION>Creator's ID for the dataset</DESCRIPTION>
</FIELD>
<FIELD ID="crdat" name="creation_date" datatype="char"
ucd="time;meta.dataset" utype="obs:DataID.Date" arraysize="*">
<DESCRIPTION>Data processing/creation date</DESCRIPTION>
</FIELD>

<FIELD ID="conty" name="dataproduct_type" datatype="char" arraysize="*"
utype="obs:Observation.ProductType">
<DESCRIPTION>Dataset content Type</DESCRIPTION>
</FIELD>
<FIELD ID="calev" name="calib_level" datatype="char" arraysize="*"
utype="obs:Observation.calibLevel">
<DESCRIPTION>Dataset content Type</DESCRIPTION>
</FIELD>
```


Figure 6: *Detailed query response using the extended list of OBS/TAP items*