International

Virtual

Observatory

Alliance

Observation Data Model Core Components and its Implementation in the Table Access Protocol

Version 1.1

IVOA Recommendation, May 09, 2017

Approved by IVOA executive committee March 20, 2017

Proposed Errata , Nov 30, 2018,// Updated Errata , May 10, 2019

Editors:

Mireille Louys, Doug Tody, Patrick Dowler, Daniel Durand

Authors:

Mireille Louys, Doug Tody, Patrick Dowler, Daniel Durand, Laurent Michel, Francois Bonnarel, Alberto Micol and the IVOA DataModel working group

Abstract

This document defines the core components of the Observation data model that are necessary to perform data discovery when querying data centers for astronomical observations of interest. It exposes use-cases to be carried out, explains the model and provides guidelines for its implementation as a data access service based on the Table Access Protocol (TAP). It aims at providing a simple model easy to understand and to implement by data providers that wish to publish their data into the Virtual Observatory. This interface integrates data modeling and data access aspects in a single service and is named ObsTAP. It will be referenced as such in the IVOA registries. In this document, the Observation Data Model Core Components (ObsCoreDM) defines the core components of queryable metadata required for global discovery of observational data. It is meant to allow a single query to be posed to TAP services at multiple sites to perform global data discovery without having to understand the details of the services present at each site. It defines a minimal set of basic metadata and thus allows for a reasonable cost of implementation by data providers. The combination of the ObsCoreDM with TAP is referred to as an ObsTAP service. As with most of the VO Data Models, ObsCoreDM makes use of STC, Utypes, Units and UCDs. The ObsCoreDM can be serialized as a VOTable. ObsCoreDM can make reference to more complete data models such as Characterisation DM, Spectrum DM or Simple Spectral Line Data Model (SSLDM).

ObsCore shares a large set of common concepts with DataSet Metadata Data Model (Cresitello-Dittmar et al. 2016) which binds together most of the data model concepts from the above models in a comprehensive and more general frame work.

This current specification on the contrary provides guidelines for implementing these concepts using the TAP protocol and answering ADQL queries. It is dedicated to global discovery.

Status of this document

This document is a revision of the ObsCore v1.0 recommendation. It extends the metadata provided for discovery of data via VO compliant TAP services. In addition, ObsCore has been selected as the core data model for data discovery by the Simple Image Access protocol version 2 (SIAv2) (Dowler, Tody et Bonnarel, IVOA Simple Image Access V2.0 2015) and future parameter-based DAL services. From the experience on the ObsCore v1.0 implementation, and to better describe datasets in support of data discovery via DAL services, new data model fields have been added.

This document has been updated by the IVOA Data Model (DM) working group, in coordination with partners involved in the definition of data access protocols (DAL) and of the ADQL language. It describes the core components and the metadata to be attached to an astronomical observation, and contains a guide for implementing this model within the Table Access Protocol (TAP) framework. Due to the DM and DAL aspects of this document, this will circulate and be reviewed by both Working Groups.

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

Acknowledgements

developing this specification and prototyping the use of ObsTAP in the VAO portal. The CANFAR project also contributed for the reference implementation of ObsTAP at CADC, Victoria, which serves a large and diverse set of data collections.

Table of Content

List of Acronyms

ADQL	Astronomical Data Query Language
ASDM	Archive Science data model : a data format for ALMA , EVLA data
DAL	Data Access Layer
DM	Data Model
FITS	Flexible Image Transport System : standard data format
ObsCore DM	Observation Core components Data Model
ObsTAP	TAP interface to ObsCore DM
TAP	Table Access Protocol
SIA	Simple Image Access
SSA	Simple Spectral Access
STC	Space-Time Coordinates
UCD	Unified Content Descriptor

1. Introduction

The first version of this model, ObsCore 1.0, originates from an initiative of the IVOA Take Up Committee that, in the course of 2009, collected a number of use cases for data discovery (see Appendix A). These use cases address the problem of an astronomer posing a world-wide query for scientific data with certain characteristics and eventually retrieving or otherwise accessing selected data products thus discovered. The ability to pose a single scientific query to multiple archives simultaneously is a fundamental use case for the Virtual Observatory. Providing a simple standard protocol such as the one described in this document increases the chances that a majority of the data providers in astronomy will be able to implement the protocol, thus allowing data discovery for almost all archived astronomical observations.

Version 1.0 and Version 1.1 of ObsCore are focused on public data. However optional fields like *obs_release_date* and *data_rights* are proposed to also support proprietary data.

The ObsCore data model is focused on describing the core metadata common to most data products distributed for astronomical observations. It is the common basis that helps to search and discover datasets across various VO compatible archives via a customized TAP protocol: ObsTAP. ObsCore also provides the core data model for discovery and description of specific types of astronomical data (e.g., images and spectra) via the "typed" VO data access protocols. These type-specific protocols may extend ObsCore to more fully describe specific types of data, but the intent is that all VO data access protocols share the same core description of the data.

In order to take into account the pixelated data such as images, data cubes, and time series as well, this version makes explicit the nature and length of the dataset axes as defined in the Characterisation data model (Louys and DataModel-WG. 2008). These allow covering the requirements for axes length (as a number of bins) expressed in added uses-cases in Appendix A, sections A.3 for data cubes, A.4 for time series, A.5 for event lists. In addition it corrects a few errors in the description of data model items found in version 1.0.

Consistency with the IVOA NDCube data model which represents N-Dimensional datasets has been improved. Therefore the main data model component of ObsCore DM, which focuses on a data product, is renamed "ObsDataset" as in 'NDCube' and 'IVOA DataSet Metadata' models, instead of '*Observation*' as named previously.

This data model does not expose the mapping of data axes to physical coordinate systems, as available for instance in FITS WCS keywords. Such information belong to the scope of the 'NDCube' and 'STCv2' data models and will be used in future versions of DAL protocols.

In the following are described the fundamental building blocks which are used to achieve the goal of *global data discoverability and accessibility*.

1.1. First building block: Data Models

Modeling of observational metadata has been an important activity of the IVOA since its creation in 2002. This modeling effort has already resulted in a number of integrated and approved IVOA standards such as the Resource Metadata, Space Time Coordinates (STC), Spectrum and SSA, and the Characterisation data models that are currently used in IVOA services and applications.

20101004
IVOA Architecture
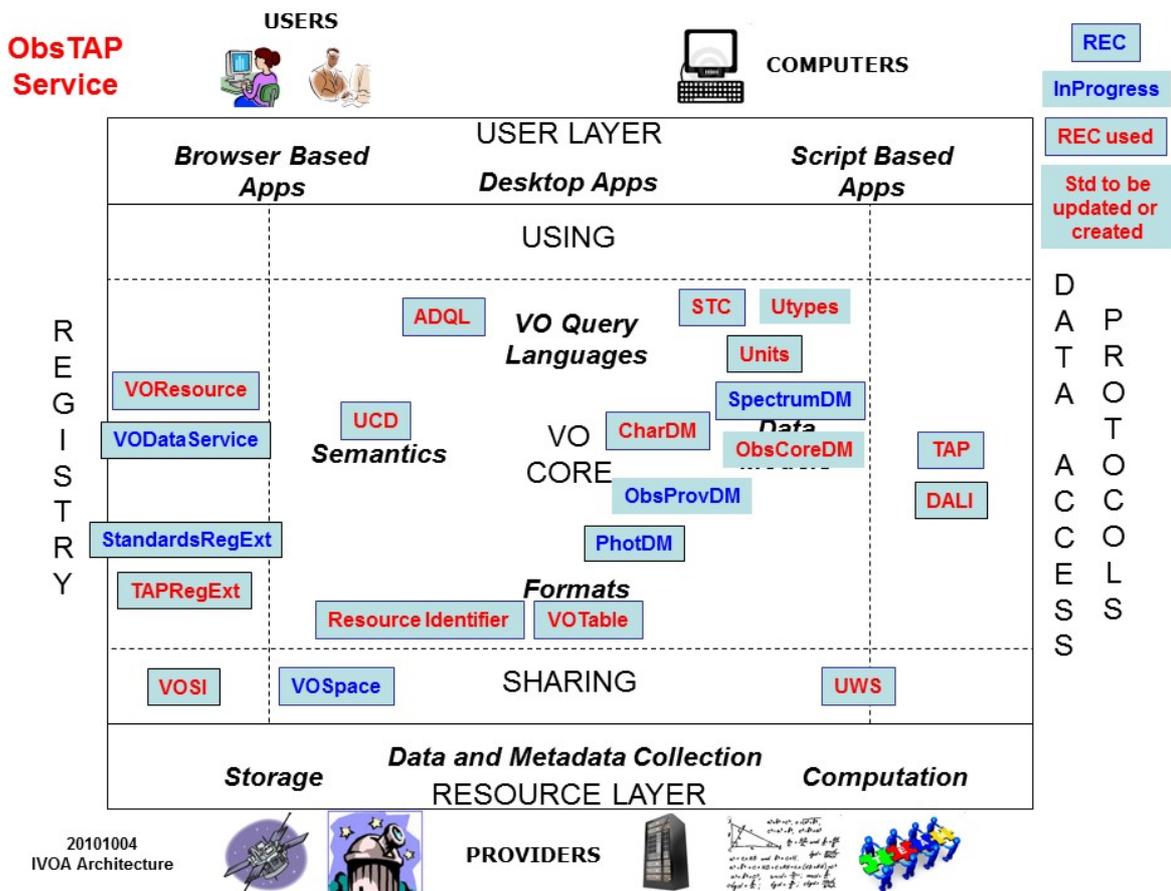
Figure 1. Architecture of an ObsTAP service: it is based on the ObsCore data model, which re-uses parts of Characterisation, Spectrum, STC data models and the UCD and Units specifications. As a service ObsTAP relies on ADQL, TAP, UWS, TAPRegExt, VOSI and VOTable. Examples and use-cases are exposed following the recommendation for DALI examples.

1.2. Second building block: the Table Access Protocol (TAP)

TAP defines a service protocol for accessing tabular data such as astronomical catalogs, or more generally, database tables. TAP allows a client to (step 1) browse through the various tables and columns (names, units, etc.) in an archive to collect the information necessary to pose a query, then (step 2) actually perform a table query. The Table Access Protocol (TAP) specification was developed and reached recommendation status in March 2010 (Dowler, Tody et Rixon, Table Access Protocol 2010).

1.3. The goal of this effort

Building on the work done on data models and TAP, it becomes possible to define a standard service protocol to expose standard metadata describing available datasets. In general, any data model can be mapped to a relational database and exposed directly with the TAP protocol. The goal of ObsTAP is to provide such a capability based upon an essential subset of the general observational data model.

Specifically, this effort aims at defining a database table to describe astronomical datasets (data products) stored in archives that can be queried directly with the TAP protocol. This is ideal for global data discovery as any type of data can be described in a straightforward and uniform fashion. The described datasets can be directly downloaded or accessed via IVOA Data Access Layer (DAL) protocols.

The final capability required to support uniform global data discovery and access, with a client sending one and the same query to multiple TAP services, is the stipulation that a uniform standard data model is exposed (through TAP) using agreed naming conventions, formats, units, and reference systems. Defining this core data model and associated query mechanism is what this document is for.

Thus the purpose of this document is twofold: (1) to define a simple data model to describe observational data, and (2) to define a standard way to expose it through the TAP protocol to provide a uniform interface to discover observational science data products of any type.

This document is organized as follows:

- Section 2 briefly presents the types of the use cases collected from the astronomical community by the IVOA Uptake committee.

- Section 3 defines the core components of the Observation data model. The elements of the data model are summarized in **Figure 2**. Mandatory ObsTAP fields are summarized in Table 1.

- Section 4 specifies the required data model fields as they are used in the TAP service: table names, column names, column data type, UCD, Utype from the Observation Core components data model, and required units.

- Section 5 describes how to register an ObsTAP service in a Virtual Observatory registry. More detailed information is available in the appendices.

- Examples are cited in Appendix A

- Section 6 summarizes updates of this document.

- Appendix A describes all the use cases as defined by the IVOA Take Up Committee.

- Appendix B contains a full description of the Observation data model Core Components.

- Appendix C shows the detailed content of the TAP_SCHEMA tables and how to build up and fill them for the implementation of an ObsTAP service.

2. Use cases

Our primary focus is on data discovery. To this end a number of use-cases have been defined, aimed at finding observational data products in the VO domain by broadcasting the same query to multiple archives (*global data discoverability and accessibility*). To achieve this we need to give data providers a set of metadata attributes that they can easily map to their database system in order to support queries of the sort listed below.

The goal is to be simple enough to be practical to implement, without attempting to exhaustively describe every particular dataset.

The main features of these use-cases are as follows:

- Support multi-wavelength as well as positional and temporal searches.

- Support any type of science data product (image, cube, spectrum, time series, instrumental data, etc.).

- Directly support the sorts of file content typically found in archives (FITS, VOTable, compressed files, instrumental data, etc.).

Further server-side processing of data is possible but is the subject of other VO protocols. More refined or advanced searches may include extra knowledge obtained by prior queries to determine the range of data products available.

The detailed list of use cases proposed for data discovery is given in Appendix A.

3. Observation Core Components Data Model

This section highlights and describes the *core components* of the Observation data model, synthetized today in the Dataset Metadata DM specification. The term "core components" is meant to refer to those elements of the larger "Observation Data Model" that are required to support the use cases listed in Appendix A. In reality this effort is the outcome of a trade-off between what astronomers want and what data providers are ready to offer. The aim is to achieve buy-in of data providers with a simple and "good enough" model to cover the majority of the use cases.

The project of elaborating a general data model for the metadata necessary to describe any astronomical observation was launched at the first Data Model WG meeting held in Cambridge, UK at the IVOA meeting in May 2003. The first Observation data model was sketched out relying on some key concepts: Dataset, Identification, Curation, physical Characterisation and Provenance (either instrumental or software). A description of the early stages of this development can be found in (Mc Dowell et al. 2005) (Observation IVOA note). Some of these concepts have already been elaborated in existing data models, namely the Spectrum data model (McDowell, Tody and al 2011) for general items such as dataset identification and curation, and the Characterisation data model (Louys and DataModel-WG. 2008) for the description of the physical axes and properties of an observation, such as coverage, resolution, sampling, and accuracy. The Core Components data model reuses the relevant elements from those models. Generalization of the observational model to support data from theoretical models (e.g., synthetic spectra) is possible but is not addressed here in order to keep the core model simple.

3.1. UML description of the model

This section provides a graphical overview of the Observation Core Components data model using the unified modeling language (UML). The UML class diagram shown in Figure 2 depicts the overall Observation Data Model, detailing those aspects that are relevant to the Core Components, while omitting those not relevant. The Characterisation classes describing how the data span along the main physical measurement axes are simplified here showing only the attributes necessary for data discovery. This is also the case for the DataID and Curation classes extracted from the Spectrum/SSA data model where only a subset of the attributes are actually necessary for data discovery. For our purposes here we show Characterisation classes only down to the level of the *Support* class (level 3).
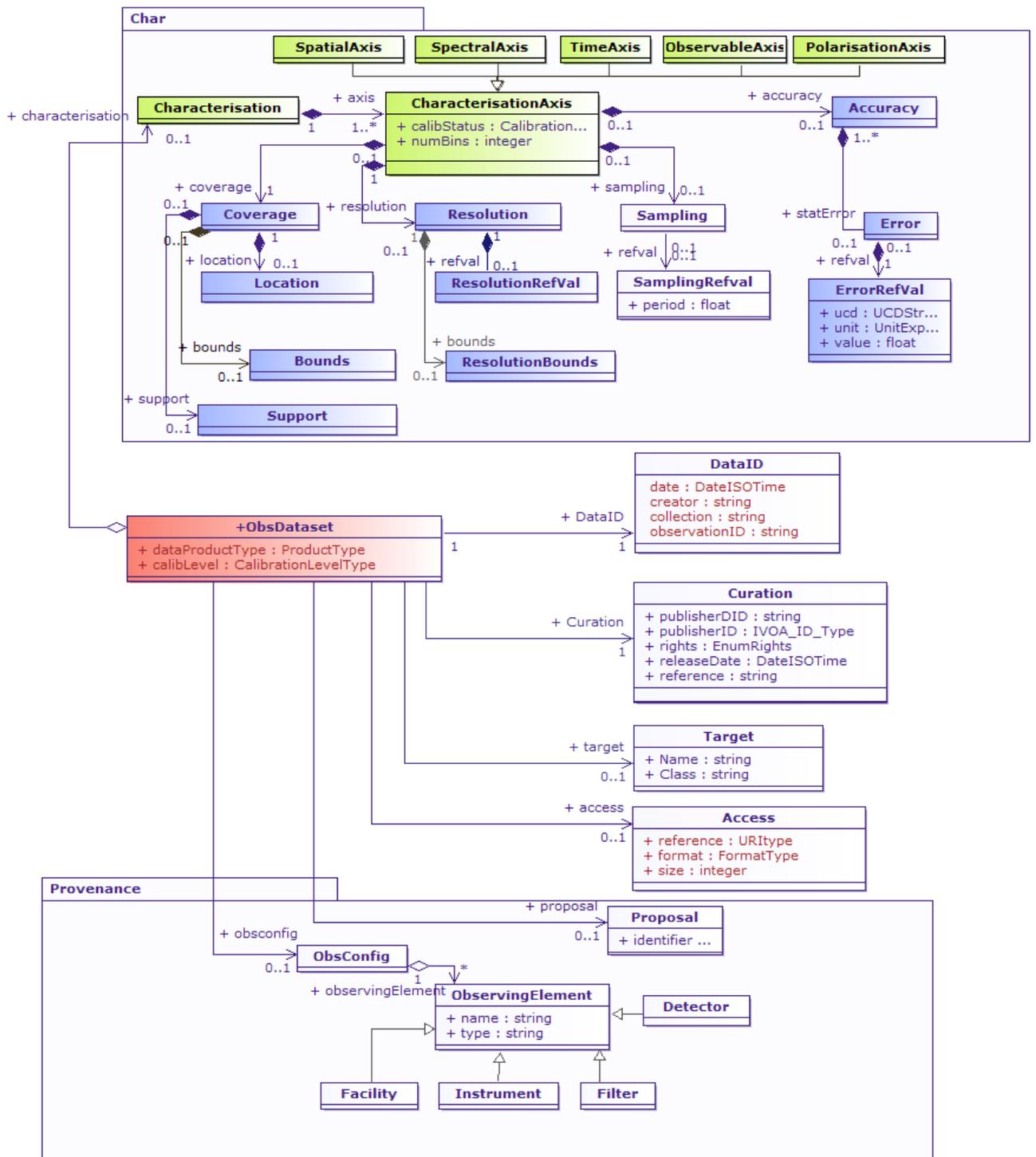
Figure 2. Depicted here are the classes used to organize observational metadata. Classes may be linked either via association or aggregation. The minimal set of necessary attributes for data discovery is shown in brown.

For the sake of clarity, the *SpatialAxis*, *SpectralAxis* and *TimeAxis* classes on the diagram are not expanded on the main class diagram. Details for these axes are shown in Figure 3 for the spatial axis, Figure 4 for the spectral axis and Figure 5 for the time axis.
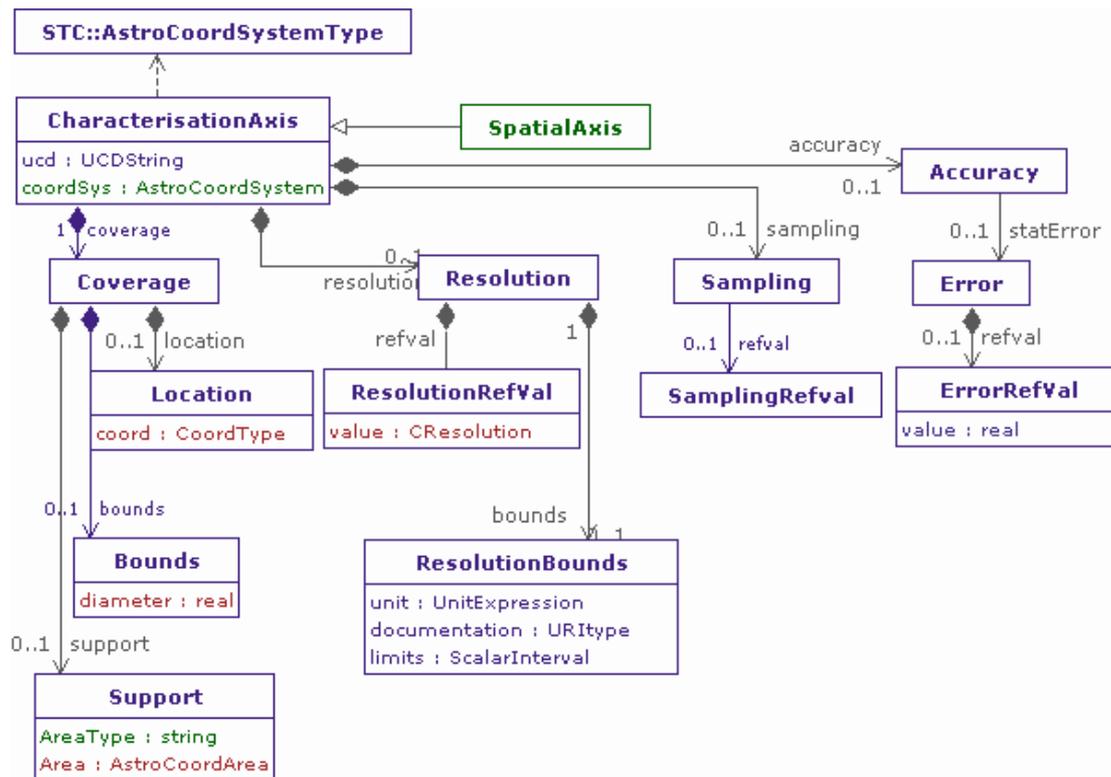
Figure 3. Details of the classes linked to the description of the spatial axis for an Observation dataset. All axes in this model inherit the main structure from the *CharacterisationAxis* class, but some peculiar attributes are necessary for Space coordinates.
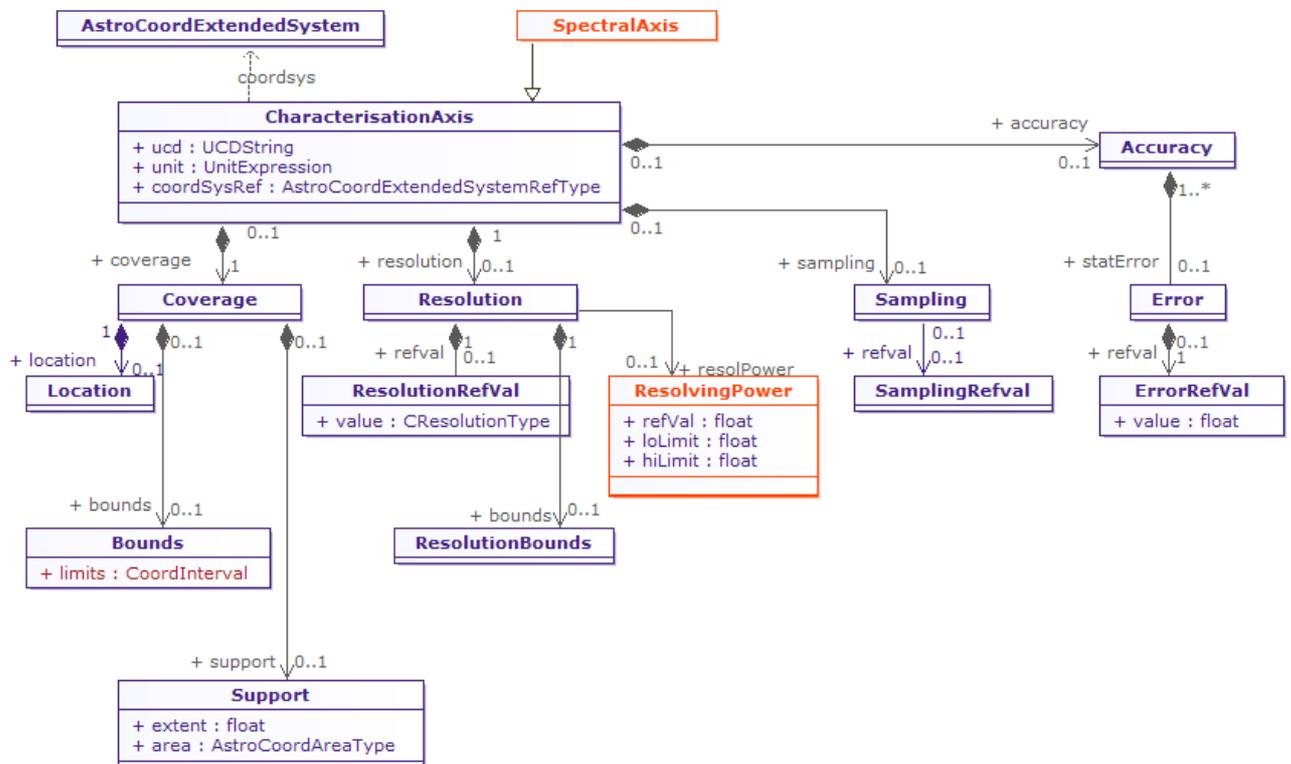
Figure 4**.** Spectral axis: details of the classes necessary to describe the spectral properties of an Observation dataset. UCD and units are essential to disentangle various possible spectral quantities.

14
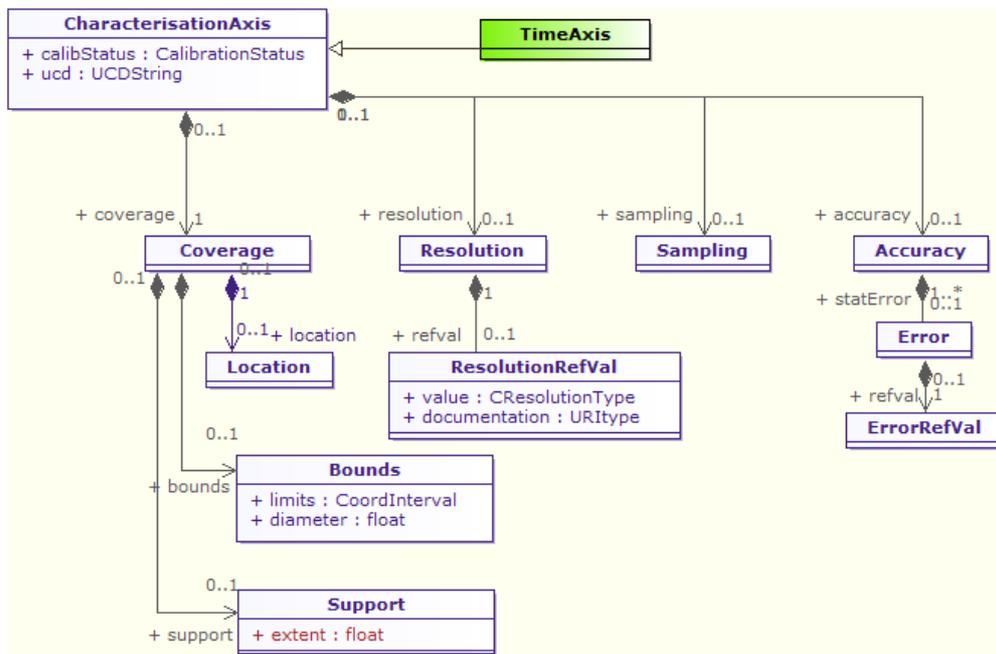
Figure 5. The classes from the Characterisation DM used to describe time metadata.

Details on the ObsCoreDM axes definitions are available in the Characterisation data model standard document (Louys and DataModel-WG. 2008). The hypertext documentation of the model is available on the IVOA web site under the ObsCore wiki page http://www.ivoa.net/internal/IVOA/ObsDMCoreComponents/ .

3.2. Main Concepts of the ObsCore Data Model

The ObsCore data model is the result of the analysis of the data discovery use cases introduced in Chapter 2. Two sets of elements have been identified: those necessary to support the provided use cases, and others that are generally useful to describe the data but are not immediately required to support the use cases. In this section only the first set is described. That set coincides with the set of parameters that any ObsTAP service **must** support. Please refer to appendix B for the detailed description of all model elements.

Table 1 lists the data model elements that any ObsTAP implementation **must** support (i.e. a column with such name must exist, though, in some cases, it could be nillable). Provision of these mandatory fields ensures that any query based on these parameters is guaranteed to be understood by all ObsTAP services.

NB: Data model fields are listed here with their **TAP column name** rather than the IVOA data model element identifiers (Utype) to ease readability. See the associated Utypes in **Appendix C**.

Column Name	Unit	Type	Description
dataproduct_type	unitless	String	Logical data product type (image etc.)
calib_level	unitless	enum integer	Calibration level {0, 1, 2, 3, 4}

obs_collection	unitless	String	Name of the data collection
obs_id	unitless	String	Observation ID
obs_publisher_did	unitless	String	Dataset identifier given by the publisher
access_url	unitless	String	URL used to access (download) dataset
access_format	unitless	String	File content format (see in App. BB.5.2)
access_estsize	kbyte	integer	Estimated size of dataset in kilo bytes
target_name	unitless	String	Astronomical object observed, if any
s_ra	deg	double	Central right ascension, ICRS
s_dec	deg	double	Central declination, ICRS
s_fov	deg	double	Diameter (bounds) of the covered region
s_region	unitless	String	Sky region covered by the data product (expressed in ICRS frame)
s_xel1	unitless	integer	Number of elements along the first spatial axis
s_xel2	unitless	integer	Number of elements along the second spatial axis
s_resolution	arcsec	double	Spatial resolution of data as FWHM
t_min	d	double	Start time in MJD
t_max	d	double	Stop time in MJD
t_exptime	s	double	Total exposure time
t_resolution	s	double	Temporal resolution
t_xel	unitless	integer	Number of elements along the time axis
em_min	m	double	Start in spectral coordinates
em_max	m	double	Stop in spectral coordinates
em_res_power	unitless	double	Spectral resolving power
em_xel	unitless	integer	Number of elements along the spectral axis
o_ucd	unitless	String	UCD of observable (e.g. phot.flux.density, phot.count, etc.)
pol_states	unitless	String	List of polarization states or NULL if not applicable
pol_xel	unitless	integer	Number of polarization samples
facility_name	unitless	String	Name of the facility used for this observation
instrument_name	unitless	String	Name of the instrument used for this observation

Table 1. Mandatory fields of the Observation Core Components data model with their name, recommended units, data type and designation.

3.3.   Specific Data Model Elements

In order to support the global data discoverability and accessibility requirements, some new concepts previously not covered by any other data model have to be introduced. This section describes those, which are: the data product type, a classification of the various levels of

calibration and processing applied to the data, the file content and format enriched and extended from the concept described in the SSA protocol (Tody, Dolensky and al. 2012). In addition, a clarification of how the terms *Observation* and *Data Product* are used in the ObsTAP context is provided as well as a discussion for composed products.

3.3.1. Data Product Type

The model defines a *data product type* attribute to describe the high level scientific classification of the data product being considered. This is coded as a string that conveys a general idea of the content and organization of a dataset. We consider a coarse classification of the types of dataset interesting for science usage, covering: image, cube, spectrum, SED, time series, visibility data, and event data.

- **image** An astronomical image, typically a 2D image with two spatial axes, e.g., a FITS image. The image content may be complex, e.g., an objective-grism observation would be considered a type of image, even though an extracted spectrum would be a Spectrum data product.

- **cube** A multidimensional astronomical image with 3 or more image axes, e.g., a spectral image cube, a polarization cube, a full Stokes radio data cube, a time image cube, etc. The most common format for astronomical "cube" data products is a multidimensional FITS image, however other formats are allowed so long as they are adequately described.

- **spectrum** Any dataset for which spectral coverage is the primary attribute, e.g., a 1D spectrum or a long slit spectrum.

- **sed** A spectral energy distribution, an advanced data product often produced by combining data from multiple observations.

- **timeseries** A one dimensional array presenting some quantity as a function of time. A light curve is a typical example of a time series dataset.

- **visibility** A visibility (radio) dataset of some sort. Typically this is instrumental data, i.e., "visibility data". A visibility dataset 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, hence can be used to produce higher level data products such as image, spectra, timeseries, and so forth.

- **event** An event-counting (e.g. X-ray or other high energy) dataset of some sort. Typically this is instrumental data, i.e., "event data". An event dataset is often a complex object containing multiple files or other substructures. An event dataset may contain data with spatial, spectral, and time information for each measured event, although the spectral resolution (energy) is sometimes limited. Event data may be used to produce higher level data products such as images or spectra.

- **measurements** A list of derived measurements gathered in a particular original dataset of one of the previous sort after some analysis processing, like a source list, or more generally a list of 'results' attached to such datasets.

  Classification of astronomical data by data product type is inherently ambiguous hence the classification scheme defined here is intentionally kept as simple as possible. The data provider should pick the primary category most appropriate for their data. Values must be specified in lower-case (in order to simplify queries). One of the defined *dataproduct_type* values **must** be used if appropriate for the data product in question, otherwise a NULL value is permitted and a more precise definition of the data product type should be given in *dataproduct_subtype*. Combination of data product types is not allowed, i.e., either one of the above values or NULL must be specified.

Further information on the specific content of a data product can be provided by the *dataproduct_subtype* data model field defined in the data model appendix B.1.2 , and by the related *obs_title* (B.3.3) and *access_format* attributes (section 4.7).

The intent of *dataproduct_type* is to provide only a general indication of the category to which the data product belongs to facilitate global data discovery.

3.3.2. Calibration level

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 **suggested classification** scale here.

Level 0: Raw instrumental data, in a proprietary or internal data-provider defined format, that needs instrument specific tools to be handled.

Level 1: Instrumental data in a standard format (FITS, VOTable, SDFITS, ASDM, etc.) which could be manipulated with standard astronomical packages.

Level 2: Calibrated, science ready data with the instrument signature removed.

Level 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.

Level 4: Analysis data products generated after some scientific data manipulation or interpretation.

The examples in the following subsection should help illustrate use of the *calib_level* attribute. It is left to the data provider to decide for ambiguous cases.

3.3.2.1. Examples of datasets and their calibration level

Here are examples of various datasets, classified according to scheme defined above.

Data product type	Data collection	Calibration Level	Comments
image	IRAS/NASA	2	Science ready data
image	IRIS/IRSA	3	Recalibrated from infrared IRAS images with removal of the sensor memory effect.
image	HDFS/ACS GOODS data	3	Image associations mosaicking/stacking
spectrum	XMM-Newton EPIC spectra	1	Raw instrumental spectrum.
cube	EVLA spectral data cube	2	Radio spectral data cube in FITS format
sed	NED SED	3	NED spectral energy distribution
event	ROSAT/HEASARC	1	Instrumental data
visibility	ALMA, Merlin, etc.	1	Instrumental data
measurements	ESO tile catalog	4	Photometric catalog of extracted sources for a tile image

timeseries	CTA reconstructed light curve	4	Reconstructed light curve following photons vs particles separation under some assumption

Table 2. Examples of datasets with their associated calibration level values.

3.3.3.   Observation and Observation Dataset

ObsTAP describes *observations in a broad sense;* exactly what comprises an "observation" is not well defined within astronomy and is left up to the data provider to define for their data. ObsTAP also describes archive *data products* (e.g., actual archive files).

The IVOA Dataset Metadata model (see http://www.ivoa.net/documents/DatasetDM/) clarifies the logical links between an Observation and an ObservationDataset i.e a data product here. It makes a distinction between an "observation" as the description of an observing experiment and its resulting datasets. Therefore the term ObsDataset is adopted in this version as a replacement of Observation in the previous ObsCore1.0 specification. It helps to handle various situations of combination, stacking, packaging of the results of performing an observation at the instrument level.

In general *an Observation Dataset,* as a result, may be composed of multiple individual data products. In this case all the data products stemming from one observation should share the same observation identifier (*obs_id*). The form of the *obs_id* string is up to the data provider so long as it uniquely identifies, within the context of the archive, all data products resulting from the observation. The individual data products associated with an observation may have different data product types, calibration levels, and so forth. ObsTAP only directly supports the description of science data products, i.e., data products which contain science data having some physical (spatial, spectral, temporal) coverage.

Two different approaches can be followed for exposing the instrumental data from an observation. One can either expose the individual science data products resulting from the observation, all sharing the same *obs_id*, or one can "package" the data products and expose the package as a single complex instrumental data product. Combinations of the two approaches are also possible, e.g., a package of all the data products, plus additional records exposing selected high priority individual data products, all sharing the same *obs_id*.

If the data products comprising an observation are exposed individually then attributes such as the calibration level can vary for different data products, e.g., the raw instrumental data as observed might be level 1, a standard pipeline data product might be level 2, and a custom user-processed data product subsequently published back to the archive might be level 3. All such data products would share the same *obs_id*.

If on the other hand all data from an observation is exposed as a single data product via ObsTAP this will likely be an aggregate of some sort (tar file, directory, etc.) containing multiple files. This latter approach is limited to instrumental data (level 0 or 1), even if objects within the aggregate observation file are higher level. From the perspective of ObsTAP this would be instrumental data, and it is up to the user or client application consuming the data to interpret the meaning of the data elements within the observation.

Which approach is best depends upon the anticipated scientific usage and is up to the data provider to determine. For example if the observational data provided is most commonly used for multi-wavelength analysis, exposing individual high level data products is likely to be the best approach. If the anticipated usage is dominated by complex analysis of instrumental data, then exposing the entire observation as a standard package of instrumental data may be preferred.

3.3.4. File Content and Format

While *dataproduct_type* specifies at a high level what a specific data product is, the *access_format* attribute specifies what is actually in the file. For example, an "image" could be a FITS image, an image embedded in a FITS multi-extension format (MEF) file, a JPEG, etc. A "spectrum" could be represented in the VO-compliant Spectrum format, or in some instrument-specific FITS binary table format. A visibility dataset could be in FITS or ASDM format, or a variety of other radio data formats. A ROSAT or Chandra observation might be presented as a 'tar' file or directory containing instrument-specific observational files. There are many such examples; we give only a few here to illustrate the concept.

Specifying the content and format of a data product is important as special software may be required to do anything useful with the data. The user needs to know exactly what the data product is before deciding to download it for analysis.

See section 4.7 for more details and implementation requirements.

4. Implementation of ObsCore in a TAP Service

The ObsCore model must be implemented within Table Access Protocol (TAP) services such that all valid queries can be executed unchanged on any service that implements the model. Additional optional or provider-defined columns are permitted (see section 4.22) so long as all mandatory columns are provided. The protocol does not specify any specific ordering of fields in the query response so long as the mandatory parameters are present in the output stream.

Here we specify an explicit mapping of the model to relational database tables; in the context of TAP this means we are specifying the logical tables as described in the TAP_SCHEMA (the TAP-required database schema where the tables and columns exposed by the service are described). This does not necessarily imply that the underlying database will have the identical structure (what is exposed through TAP could be, for example, a database view of the underlying database tables), but in most cases the relationship between TAP_SCHEMA description and the underlying tables is straightforward.

schema_name	table_name	*Description*
ivoa	ivoa.ObsCore	ObsCore 1.1

Table 3. TAP_SCHEMA.tables for implementation of the ObsCore model.

table_name	column_name	*data type*	*units*	*Constraint*
ivoa.ObsCore	dataproduct_type	adql:VARCHAR		
ivoa.ObsCore	calib_level	adql:INTEGER		not null
ivoa.ObsCore	obs_collection	adql:VARCHAR		not null
ivoa.ObsCore	obs_id	adql:VARCHAR		not null
ivoa.ObsCore	obs_publisher_did	adql:VARCHAR		not null
ivoa.ObsCore	access_url	adql:CLOB		
ivoa.ObsCore	access_format	adql:VARCHAR		
ivoa.ObsCore	access_estsize	adql:BIGINT	kbyte	
ivoa.ObsCore	target_name	adql:VARCHAR		
ivoa.ObsCore	s_ra	adql:DOUBLE	deg	

ivoa.ObsCore	s_dec	adql:DOUBLE	deg	
ivoa.ObsCore	s_fov	adql:DOUBLE	deg	
ivoa.ObsCore	s_region	adql:REGION		
ivoa.ObsCore	s_resolution	adql:DOUBLE	arcsec	
ivoa.ObsCore	s_xel1	adql:BIGINT		
ivoa.ObsCore	s_xel2	adql:BIGINT		
ivoa.ObsCore	t_min	adql:DOUBLE	d	
ivoa.ObsCore	t_max	adql:DOUBLE	d	
ivoa.ObsCore	t_exptime	adql:DOUBLE	s	
ivoa.ObsCore	t_resolution	adql:DOUBLE	s	
ivoa.ObsCore	t_xel	adql:BIGINT		
ivoa.ObsCore	em_min	adql:DOUBLE	m	
ivoa.ObsCore	em_max	adql:DOUBLE	m	
ivoa.ObsCore	em_res_power	adql:DOUBLE		
ivoa.ObsCore	em_xel	adql:BIGINT		
ivoa.ObsCore	o_ucd	adql:VARCHAR		
ivoa.ObsCore	pol_states	adql:VARCHAR		
ivoa.ObsCore	pol_xel	adql:BIGINT		
ivoa.ObsCore	facility_name	adql:VARCHAR		
ivoa.ObsCore	instrument_name	adql:VARCHAR		

Table 4. List of the minimal set of data model fields to implement for an ObsTAP service. See tables of Appendix C for the full description of the TAP_SCHEMA.columns table.

Table 3 and Table 4 provide the primary information needed to describe the ObsCore model in terms of `TAP_SCHEMA` tables and columns.

The "datatype" column values should follow the TAP standard specification and are bound to the TAP specification version used to implement the model. Here what is shown applies for TAP v1.0.

The content of the "constraint" column specified in Table 4 above is not part of the `TAP_SCHEMA.columns` description, but is required by the ObsCore model and specified here to make this clear to implementers. Additional standard content for the individual columns is specified below.

4.1. Data Product Type (dataproduct_type)

The *dataproduct_type* column contains a simple string value describing the primary nature of the data product. It should assume one of these string values: **image**, **cube**, **spectrum**, **sed**, **timeseries**, **visibility**, **event or measurements**. These values are described in section 3.3.1. A NULL value is permitted, but only in the event that none of the proposed values can be used to describe the dataset. The optional field *dataproduct_subtype* (B.1.2) may be used to more precisely define the nature of the dataset. Values in the *dataproduct_type* column **must** be written in lower case. Specifying this field along with the desired spatial and spectral coverage will be enough to discover data of interest in many common cases.

Usage: select * from ivoa.ObsCore where dataproduct_type='image' returns only image data.

1.1.1. Caveat while using dataproduct_type="measurements"

Note that "measurements" extends the set of accepted values for **dataproduct_type** in ObsCore 1.0. This extension is meant to expose derived data products **together** with the progenitor observation dataset.

A few mandatory keywords for the axes description may be non-applicable for such a data product. In this case the coverage on spatial, energy, time, and polarization may inherit the values from the ObsCore description of its progenitor.

Progenitors and their derived data products **must** have the same *obs_id*.

4.2.   Calibration Level (calib_level)

The *calib_level* column tells the user the amount of calibration processing that has been applied to create the data product. *calib_level* **must** assume one value among {0, 1, 2 ,3, 4}. Please refer to section 3.3.2 for a full description of the various categories. Data providers decide which value best describes their data products.

Values in the *calib_level* column **must not** be NULL.

Query usage: "`select * from ivoa.ObsCore where calib_level >2`" returns enhanced data products.

4.3.   Collection Name (obs_collection)

The *obs_collection* column identifies the data collection to which the data product belongs. A *data collection* is any logical collection of datasets which are alike in some fashion. Typical data collections might be all the data from a particular telescope, instrument, or survey. The value is either the registered shortname for the data collection, the full registered IVOA identifier for the collection, or a data provider defined shortname for the collection. Often the collection name will be set to the name of the instrument that generated the data. In that case we suggest specifying the collection name as a string composed of the facility name, followed by a slash, followed by the instrument name.

> Examples : HST/WFPC2, VLT/FORS2, CHANDRA/ACIS-S.

There are other cases where it makes no sense to use the instrument name, may be because the data product used data from different instruments or facilities, or for other reasons. Examples: SDSS-DR7, etc.

In practice this is not a very precisely defined field. What is important is for the data provider to use a collection name which is familiar to astronomers and discriminative to point easily on datasets of interest.

Values in the *obs_collection* column **must not** be NULL.

4.4.   Observation Identifier (obs_id)

The *obs_id* column defines a unique identifier for an observation as explained in section 3.3.3 . In the case where multiple data products are available for an observation (e.g. with different calibration levels), the *obs_id* value will be the same for each data product comprising the observation. This is equivalent to the dataset name for many archives where dataset name could have many files associated with them. The returned *obs_id* for an archival observation should remain identical through time for future reference.

In the case of some advanced data products (with *calib_level* ≥ 3), the data product may be the result of combining data from multiple primary (physical) observations. In this case the resulting data product is a new processed "observation" to which a new unique observation identifier should be assigned. If the advanced processing results in several associated data

products they should share the same *obs_id*. Describing the provenance of such an advanced data product is possible, but is out of scope for ObsTAP.

Values in the *obs_id* column **must not** be NULL.

4.5.   Publisher Dataset Identifier (obs_publisher_did)

The *obs_publisher_did* column contains the IVOA dataset identifier (Plante and al. 2007) for the published data product. **This value must be unique within the namespace controlled by the dataset publisher** (data center). The value will also be globally unique since each publisher has a unique IVOA registered publisher ID. The same dataset may however have more than one publisher dataset identifier if it is published in more than one location; the creator DID, if defined for the given dataset, would be the same regardless of where the data is published.

The returned *obs_publisher_did* for a static data product should remain identical through time for future reference.

Values in the *obs_publisher_did* column **must not** be NULL.

4.6.   Access URL (access_url)

The *access_url* column contains a URL that can be used to download the data product (as a file of some sort).

We specify the data type as CLOB (character large object) in the TAP service so that users will know they can only use the *access_url* column in the SELECT clause of a query. That is, users cannot specify this column as part of a condition in the WHERE clause and implementers are free to generate the URL on the fly during output (rather than being forced to store it statically in the database).

More details are given on the use of CLOB data types for the TAP SCHEMA in the TAP Standard document (Dowler, Tody et Rixon, Table Access Protocol 2010), section 2.5 Table upload.

Access URLs are not guaranteed to remain valid and unchanged indefinitely. To access a specific data product after a period of time (e.g., days or weeks) a query should be performed (e.g., using *obs_publisher_did*) to obtain a fresh access URL.

4.7.   Access Format (access_format)

The *access_format* column specifies the format of the data product if downloaded as a file. This data model field is important both for data discovery and for the client to evaluate whether it will be able to actually use the data product if downloaded.

MIME types are often used to specify file formats in existing protocols such as HTTP (Authority Internet Assigned Numbers 2007). However when dealing with astronomical observations as in ObsTAP services, more information about the format of the data is required than can be specified by conventional MIME types. For instance we might want to distinguish between various formats like multi-extension FITS (e.g. for CCD mosaic instruments or MUSE IFU data), or ASDM (e.g. for ALMA or other interferometry observations). Even for something as fundamental to astronomy as FITS binary table there is currently no standardized MIME type other than the generic `application/FITS`.

While standard MIME types are limited when it comes to describing the many data formats actually in use within astronomy, they are ideal for specifying common file types such as HTML and XML, the various graphics file types, text, PDF, and so forth, all of which can be used to describe aspects of observational data. Furthermore the MIME type scheme is extensible, allowing new formats which are not yet standardized to be specified. Hence what we propose

here is to adopt the MIME type mechanism to describe the *file format* of a science data product, defining new custom types as needed. Note this is distinct from the *science content* which is specified by the data product type and subtype. The same content can potentially be represented in multiple formats hence these are distinct.

The following table illustrates some common astronomical file formats. This list is by no means intended to be comprehensive; rather it illustrates the approach while defining standard values for some common formats. Some randomly selected formats are included to illustrate the approach. We can extend this list as experience is gained using ObsTAP to describe actual data archives.

MIME-type	*Shortname*	*Definition*
image/fits	fits	Any multidimensional regularly sampled FITS image or cube
image/jpeg	jpeg	A 2D JPEG graphic image (likewise for GIF, PNG, etc.)
application/fits	fits	Any generic FITS file
application/x-fits-bintable	bintable	A FITS binary table (single BINTABLE extension)
application/x-fits-mef	mef	A FITS multi-extension file (multiple extensions)
application/x-fits-uvfits	uvfits	A FITS file in UVFITS format (likewise SDFITS etc.)
application/x-fits-euro3d	euro3d	A FITS file in Euro3D format (multiobject spectroscopy)
application/x-votable+xml	VOTable	Any generic VOTable file
application/x-asdm	asdm	ALMA science data model (final export format still TBD)
application/pdf	pdf	Any PDF file
text/html	html	Text in HTML format
text/xml	xml	Any generic XML file
text/plain	txt	Any generic text file
text/csv	csv	Tabular data in comma separated values format
text/tab-separated-values	tsv	Tabular data in tab separated values format
application/x-tar	tar	Multiple files archive in TAR format
application/zip	zip	Multiple files archive in ZIP format
application/x-directory	dir	Multiple files archive returned as a text list
image/x-fits-gzip	fits	A GZIP-compressed FITS image
image/x-fits-hcompress	fits	A FITS image using HCOMPRESS compression
application/x-tar-gzip	gtar	A GZIP-compressed TAR file (x-gtar also sometimes used)
application/x-votable+xml; content=datalink	datalink	A datalink response containing links to data sets or services attached to the current dataset

The value of *access_format* should be a MIME type, either a standard MIME type, an extended MIME type from the above table, or a new custom MIME type defined by the data provider. The short names suggested here are not used directly by *access_format.*

Custom file formats should be specified using a MIME type such as "*application/x-<whatever>*". This can be used for any file format including custom binary file formats.

Observational datasets consisting of multiple instrument-specific files may be exposed in formats like application/x-directory, application/x-tar or application/x-tar-gzip. Details of the package content and how to access inner data products are described in the IVOA Data Link specification (Dowler, Bonnarel, et al. 2015). See the example presented in section C.1.1 .

Compression is inherent in some file formats, e.g., ZIP or JPEG. In other formats it is optional and is indicated by having multiple versions of the format, e.g. *image/fits* or *image/x-fits-gzip* for a GZIP-compressed FITS image (the "x-" prefix is required for anything which is not a registered standard MIME type).

The *access_url* field may also point to a datalink service. This is stipulated by the '*application/x-votable+xml;content=datalink*' access format.

This service will return a response containing attached files related to the discovered dataset (previews, tar ball…). It can also contain descriptions of services running operations on the dataset like cut-outs..

4.8. Estimated Download Size (access_estsize)

The *access_estsize* column contains the approximate size (in kilobytes) of the file available via the *access_url*. This is used only to gain some idea of the size of a data product before downloading it, hence only an approximate value is required. Provision of dataset size estimates is important whenever it is possible that datasets can be very large.

4.9. Target Name (target_name)

The *target_name* column contains the name of the target of the observation, if any. This is typically the name of an astronomical object, but could be the name of a survey field.

The target name is most useful for output, to identify the target of an observation to the user. In queries it is generally better to refer to astronomical objects by position, using a name resolver to convert the target name into a coordinate (when possible).

4.10. Central Coordinates (s_ra, s_dec)

The coordinate system in which coordinates are expressed is ICRS. The *s_ra* column specifies the ICRS Right Ascension of the center of the observation. The *s_dec* column specifies the ICRS Declination of the center of the observation.

4.11. Spatial Extent (s_fov)

The *s_fov* column (1D size of the field of view) contains the approximate size of the region covered by the data product. For a circular region, this is the diameter (not the radius). For most data products the value given should be large enough to include the entire area of the observation; coverage within the bounded region need not be complete, for example if the specified FOV encompasses a rotated rectangular region. For observations which do not have a well-defined boundary, e.g. radio or high energy observations, a characteristic value should be given.

The *s_fov* attribute provides a simple way to characterize and use (e.g. for discovery computations) the approximate spatial coverage of a data product. The spatial coverage of a data product can be more precisely specified using the *s_region* attribute (4.12).

4.12. Spatial Coverage (s_region)

The *s_region* column can be used to precisely specify the covered spatial region of a data product.

It is often an exact, or almost exact, representation of the illumination region of a given observation defined in a standard way by the concept of Support in the Characterisation data model.

We specify the data type as adql:VARCHAR so that users can specify spatial queries using a single column and in a limited number of ways. If implemented in TAP 1.0, and included in the select list of the query, the output is always an STC-S string as described in (Dowler, Tody et Rixon, Table Access Protocol 2010) [section 6]. In the WHERE clause, the *s_region* column can be used with the ADQL geometry functions (INTERSECTS, CONTAINS) to specify conditions; the service will generally have to translate these into native SQL that enforces the same conditions or a suitable approximation. Implementers may approximate the spatial query conditions by translating the INTERSECTS and CONTAINS function calls in the query.

Because ObsTAP relies on ADQL queries and builds up on TAP, the mapping between the ObsCore model data types, as shown in Table 1. Mandatory fields of the Observation Core Components data model with their name, recommended units, data type and designation.should be adjusted to the definitions stated in the TAP version used for the ObsTAP service.

Region computations are an advanced query capability which may not be supported by all services. Services should however specify *s_region* when possible to more precisely specify the spatial coverage of an observation.

4.13. Spatial Resolution (s_resolution)

The *s_resolution* column specifies a reference value chosen by the data provider for the estimated spatial resolution of the data product in arcseconds. This refers to the smallest spatial feature in the observed signal that can be resolved.

In cases where the spatial resolution varies across the field the best spatial resolution (smallest resolvable spatial feature) should be specified. In cases where the spatial frequency sampling of an observation is complex (e.g., interferometry) a typical value for spatial resolution estimate should be given; additional characterisation may be necessary to fully specify the spatial characteristics of the data.

4.14. Time Bounds (t_min, t_max)

The *t_min* column contains the start time of the observation specified in MJD. The *t_max* column contains the stop time of the observation specified in MJD. In case of data products result of the combination of multiple frames, *t_min* must be the minimum of the start times, and *t_max* as the maximum of the stop times.

4.15. Exposure Time (t_exptime)

The *t_exptime* column contains the exposure time. For simple exposures, this is just *t_max* - *t_min* expressed in seconds. For data where the detector is not active at all times (e.g. data products made by combining exposures taken at different times), the *t_exptime* will be smaller than *t_max* - *t_min*. For data where the *t_exptime* is not constant over the entire data product, the median exposure time per pixel is a good way to characterize the typical value. In some cases, *t_exptime* is generally used as an indicator of the relative sensitivity (depth) *within a single data collection* (e.g. *obs_collection*); data providers should supply a suitable relative value when it is not feasible to define or compute the true exposure time.

In case of targeted observations, on the contrary the exposure time is often adjusted to achieve similar signal to noise ratio for different targets.

4.16. Time Resolution (t_resolution)

The *t_resolution* column is the minimal interpretable interval between two points along the time axis. This can be an average or representative value. For products with no sampling along the time axis, the *t_resolution* could be set to the exposure time or could be null. If set to exposure time, one could compose a WHERE clause like: `WHERE t_resolution < t_exptime` to find those products which are time resolved.

This implementation preference avoids dealing with undefined data model fields as originally considered in the Characterisation data model for unresolved time axis so NULL value is preferred to not defined.

4.17. Spectral Bounds (em_min, em_max)

The *em_min* column specifies the minimum spectral value observed, expressed as a vacuum wavelength in meters.

The *em_max* column contains the maximum spectral value observed, expressed as a vacuum wavelength in meters.

As mentioned in the data model in Appendix B, at least 3 physical quantities could in principle be used to represent the spectral axis: energy, wavelength or frequency; which is most appropriate depends upon the observation domain. For ObsTAP we are less concerned with how to present data to the user than with providing a simple and uniform way to describe astronomical data, hence we restrict the spectral bounds units to wavelength in meters in vacuum. Conversion to other quantities could be performed either on the client side for an application encapsulating queries, and/or on the server side, for a data provider to expose its data from other regimes to ObsTAP queries.

4.18. Spectral Resolving Power (em_res_power)

The *em_res_power* column contains the typical or characteristic spectral resolving power defined as $\lambda/\delta\lambda$. The value is dimensionless.

4.19. Polarization states (pol_states)

Polarisation states can also be described as a simple list of values if the dataset contains polarization data.

See section   BB.6.5 for details.

4.20. Observable Axis Description (o_ucd)

The *o_ucd* column specifies a UCD (Preite Martinez, et al. 2007) describing the nature of the observable within the data product. The observable is the measured quantity, for a sampled dataset, such as for example photon counts or flux density stored in the pixel value within an image. Often for optical astronomical images the value would be `phot.count`; for fully flux calibrated data a value such as `phot.flux.density` (usually specified in Jy) would be used. Any valid UCD is permitted. If no appropriate UCD is defined the field should be left NULL (the IVOA provides a process by which new UCDs can be defined).

In the case of event lists all components could be considered as observables prior to sampling, then *o_ucd* must be left NULL, unless the data provider wants to highlight a specific axis like `phot.count`.

4.21. Axes lengths (s_xel1, s_xel2, em_xel, t_xel, pol_xel)

The lengths of each data axis (spatial, spectral, time, polarization) defined as a number of elements along each of these axes are included in this specification for ObsCore v1.1. This data model element was already defined as an attribute of the CharacterisationAxis Class in the Characterisation Data model. This provides quantitative information on the geometry of the data portion along the axes defined in the Characterisation Data Model. Various use-cases in Appendix A, section A.6 illustrate these discovery scenarios.

- s_xel1, s_xel2 specify the number of values spanned along the spatial dimensions
- em_xel, t_xel specify the number of values spanned for the spectral and time axis respectively.
- pol_xel specifies the number of polarization states present in the dataset

This information helps to plan data selection, data slicing or sub setting following data discovery and will be used for building up extracted subsets on the fly.

For pixelated data this concept clearly represents the number of samples along each axis.

In the case of non-pixelated data, like event lists, where several events can be gathered in one time bin or energy bin for instance, these attributes should be set to -1. The number of elements in such lists is a different property and should be represented in the NDCube DM which tackles sparse data sets.

4.22. Additional Columns

Service providers may include additional columns in the *ivoa.ObsCore* table to expose additional metadata. These columns must be described in the *TAP_SCHEMA.columns* table and in the output from the VOSI-tables resource ([VOSI] Grid and Web service WG 2010). Users may access these columns by examining the column metadata for individual services and then using them explicitly in queries or by selecting all columns in the query (e.g. "select * from ivoa.ObsCore ..." in an ADQL query). In order to provide homogeneity in the keywords used as optional fields, we recommend where possible to use the items defined in the full data model (Appendix B) and flagged as optional. ObsTAP compliant services will support all columns defined as mandatory and possibly some of the optional ones. Queries built up using additional columns defined specifically for a given archive might not be portable.

5. Registering an ObsTAP Service

The standard identifier for this version of the ObsCore model now follows the IVOA Identifiers v2.0 specification (Demleitner, Plante et Noddle, IVOA Identifiers v2.0 Recommendation 2016) and should be *ivo://ivoa.net/std/ObsCore#core-1.1*

The ObsCore data model will be registered using this identifier in the StandardsRegExt (standards registry extension) definition.

TAP services that implement the ObsCore model should be registered to indicate this fact so that users can easily find all services that accept ObsCore queries. This can be done in any registry by using the keyword "ObsCore" to describe the service. In addition, fine-grained registries may include the complete VODataService table set description.

The TAPRegExt (Table Access Protocol registry extension) (Demleitner, Dowler, et al. 2012) provides a mechanism (the '**dataModel**' element) to list one or more data models that are supported by a TAP service. The data model support uses the ivo standard identifier (above). One or more '**dataModel**' elements may be included as child elements of the

'**capability'** element describing the TAP service which is the '**capability'** element with the following attributes:

standardID="ivo://ivoa.net/std/TAP"(or later version)

xmlns:tr="http://www.ivoa.net/xml/TAPRegExt/v1.0" (or later version)

xsi:type="tr:TableAccess"

For TAP services that support ObsCore-1.0 only, the 'dataModel' element would be:

<*dataModel* *ivo-id*=*"ivo://ivoa.net/std/ObsCore/v1.0">ObsCore-1.0</dataModel*>

as defined in the early version 1.0 of this specification.

TAP services that support ObsCore-1.1 with the dimensions elements s_xel1, s_xel2, em_xel, etc. and utypes updates should include instead:

<*dataModel* *ivo-id*=*"ivo://ivoa.net/std/ObsCore#core-1.1">ObsCore-1.1</dataModel*>

In general, the data model support in TAPRegExt can be used when a TAP service contains tables and columns described with Utypes from a standard data model; it is not generally necessary to have all the Utypes (e.g. the complete model). However, since the ObsCore data model is a physical model designed specifically to be implemented in TAP services, the standard identifier must only be used to specify data model support in the TAPRegExt if the *ivoa.ObsCore* table is available and contains all the mandatory columns[1].

6. Changes from Earlier Versions

Errata on REC 2017 May 09:

Removed FWHM from Time resolution definition in Table 1. /2018/05/12

Updated UCD column for *obs_publisher_did* and *publisher_id* to `meta.ref.ivoid`

Version REC 2017 May 09:   (after final and careful read from A. Micol)

- corrected erroneous section number 7 into 6,

- corrected datamodel identifier in section 5,

- corrected ucd `meta.ref.url` in obs_publisher_did and publisher_id   as `meta.ref.uri` instead of url,

- reword *t_resolution* paragraph in section .

.

Version PR 1.1 2017: Mar 05 2017: table 4 in Appendix B: definition of *pol_states* and definition of *o_calib_status*

Version PR 1.1 Oct04: typos and corrections of xml document example in App. C section 3

---

[1] Additional columns with optional ObsCore Utypes, Utypes from other data models, or no Utypes at all are allowed

Version PR 1.1 March 2016 to PR1.1 September 2016:

- In TAP_SCHEMA table 3 changed datatype for access_estsize to adql:BIGINT instead of adql:INTEGER and for pol_xel to adql:INTEGER instead of ADQL:BIGINT

- In Abstract: Clarified links with DataSet Metadata DM and emphasizes the TAP/ADQL implementation aspects of this specification

- Define 'measurements' as a new dataproduct_type for source lists, catalogs, and products derived from data extraction/interpretation.

- Extend allowed values for calib_level up to 4, to represent data products obtained as results of deep analysis processing.

- Cleanup for expressing Use-cases in Appendix A

- IVOA IDs in TAPRegExt updated to comply to IVOA identifiers version 2.0

- Typo corrections from reviewers 'comments listed on the RFC page

- TAPRegExt section 5 rewrite

- Use case appendix A: reformulate request for service capabilities

Version 1.0 to 1.1 March 2016:

- Homogenize case in Utypes strings

- Improve ucd tags for *s_region* now labeled with `pos.outline;obs.field` instead of `phys.area;obsfield`

- Include axes dimensions (number of elements along one axis) expressed as s_xel, em_xel, t_xel, etc , as an extrapolation of the definition for pi_xel, vo_xel , etc.

- Insert field *s_pixel_scale* that was missing in Appendix B Table 5 and App.C Table 7.

- Homogenize root class name: Obs(ervation) changed to ObsDataset according to Dataset Metadata data model and Cube DM

- Enlarge the possible values for ***em_ucd*** to allow the search for Doppler features (velocity cubes)

- Correct minor Utype typos and inconsistencies.

References

[VOSI] Grid and Web service WG. "IVOA Support Interfaces." Edited by Matthew Graham and Guy Rixon. *http://www.ivoa.net/Documents/VOSI/index.html.* IVOA Standards, 2010.

Authority Internet Assigned Numbers. "MIME Media Types." 2007.

Bonnarel, F., I. Chilingarian, and M. Louys. "Characterisation Data model v2.0: variability and complex observations." IVOA Standards, (in prep.).

Cresitello-Dittmar, Mark., and al. "IVOA Dataset Metadata Model." *ivoa.net.* in preparation 2016. http://www.ivoa.net/documents/DatasetDM/20160317/index.html.

Demleitner, Markus, Patrick Dowler, Ray Plante, Guy Rixon, and Mark Taylor. "TAPRegExt: a VOResource Schema Extension for Describing TAP Services Version1.1." *http://www.ivoa.net/Documents/TAPRegExt/index.html.* IVOA Standards, 2012.

Demleitner, Markus, Ray Plante, and Keith Noddle. "IVOA Identifiers v2.0 Recommendation ." May 23, 2016. http://www.ivoa.net/documents/IVOAIdentifiers/20160523/REC-Identifiers-2.0.html.

Derrière, Sebastien. "Providing Photometric data measurements description in VOTable v0.1." *http://ivoa.net/Documents.* IVOA Notes, 2010.

Dowler, P., D . Tody, and F. Bonnarel. "IVOA Simple Image Access V2.0." Edited by IVOA Standards. *IVOA Standard*, 2015.

Dowler, P., D. Tody, and G. Rixon. "Table Access Protocol." *http://www.ivoa.net/Documents/TAP/20100327/REC-TAP-1.0.pdf.* IVOA Standards, 2010.

Dowler, P., Demleitner,M. , Mark Taylor, M. , Tody,D. "Data Access Layer Interface - IVOA Recommendation." 2013.

Dowler, P., F. Bonnarel, L. Michel, and M. Demleitner. "IVOA DataLink." *http://www.ivoa.net/documents/DataLink/.* IVOA Standards, 2015.

Greisen, E. W., and M.R. Calabretta. "Representations of world coordinates in FITS." *Astronomy & Astrophysics* 395 (2002): 1061-1075.

Hanisch, R. "Resource Metadata for the Virtual Observatory." Edited by IVOA Standards. http://www.ivoa.net/Documents/latest/RM.html, 2007.

Louys, M, and F. Bonnarel. "List of UCD for observable quantities." IVOA wiki site, 2010.

Louys, M., and DataModel-WG. "Data Model for Astronomical DataSet Characterisation." Edited by Mireille Louys, Anita Richards, François Bonnarel, Jonathan McDowell and al. *http://www.ivoa.net/Documents/latest/CharacterisationDM.html.* IVOA Standards, 2008.

Mc Dowell, Jonathan, and al. "Data Model for Observation." Edited by Jonathan McDowel, François Bonnarel, David David Giaretta, Gerard Lemson, Mireille Louys and Alberto Micol. *http://www.ivoa.net/Documents/Notes/DMObs/DMObs-20050421.pdf.* IVOA Note, 2005.

McDowell, Jonathan, Doug Tody, and al. "IVOA Spectral Data Model v1.1." *http://www.ivoa.net/Documents//SpectrumDM/20110318/.* IVOA Standards, 2011.

Plante, R., and al. "IVOA Identifiers v1.12." IVOA Standards, 2007.

Plante, Ray, and al. "VODataService : a VOResource schema extension for describing collections and services v 1.1." *http://www.ivoa.net/Documents/latest/VODataService/.* IVOA Standards, 2010.

Preite Martinez, A., S. Derriere, N. Delmotte, N. Gray, and al. "The UCD1+ controlled vocabulary." *http://www.ivoa.net/Documents/latest/UCDlist.html.* IVOA standards, 2007.

Richards, A., and F. Bonnarel. "Note on the description of polarisation data v1.0." *http://www.ivoa.net/Documents/Notes/Polarization/.* IVOA Note, 2010.

Rots, Arnold. "Space-Time Coordinate Metadata for the Virtual Observatory." *http://www.ivoa.net/Documents/REC/DM/STC-20071030.pdf.* IVOA Standards, 2007.

Tody, D., M. Dolensky, and al. "Simple Spectral Access Protocol 1.1." Edited by D. Tody and M. Dolensky. *http://www.ivoa.net/Documents/REC/DAL/SSA-20110417.pdf.* IVOA Standards, 2012.

Appendix A: Use Cases in detail

The ability to discover data of a certain kind (images, spectra, cubes, event lists) according to scientific criteria (e.g., a given sky position, spectral coverage including spectral line X, spatial resolution better than Y, resolving power greater than Z) is central to archival astronomy. A special Take Up Committee of the IVOA was formed in 2009 to stimulate IVOA work in the area of catalog-based science data access to allow astronomers to easily query and access scientific data. This committee came up with a list of data discovery use cases expressed as a set of constraints on selected scientific parameters to be used to query for datasets of interest. The full list of use cases is summarized below.

Please note that for most science cases, a full TAP implementation is required for them to work as well as STC regions support. (Rots 2007)

Some of the use-cases listed by the committee require advanced functionalities like "search by type", "query from an input list", and have not been fully developed here.

Typical use-cases are described below.

A wider set of working examples (beta release), is available at http://saada.unistra.fr/voexamples/show/ObsCore a DALI compliant example service developed by Laurent Michel, Mireille Louys and Daniel Durand (May 2016, in progress).

Simple Examples

Simple Query by Position

Show me a list of all data that satisfies:

    I.   Datatype=any
    II. 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
CONTAINS(POINT('ICRS',16.0,40.0),s_region)=1
```

This query could be submitted to a remote TAP service using the *curl* application as follows (in this example a CADC TAP service is referenced):

```
curl -v -L -d "REQUEST=doQuery&LANG=ADQL&QUERY=select * from
ivoa.ObsCore where CONTAINS(POINT('ICRS',16.0,40.0 ),s_region)=1"
"http://www.cadc.hia.nrc.gc.ca/caom/sync"
```

More constraints are added in the following use-case (1.3).

Query Images by both Spatial and Spectral Attributes

Show me a list of all data that satisfies:

    I.   DataType=Image
    II. Spatial resolution better than 0.3 arc seconds
    III. Filter = J or H or K
    IV. RA between 16 hours and 17 hours
    V. DEC between 10 degrees and 11 degrees

Such a query needs to compute RA in degrees, extract information from Filter and adjust spectral intervals for search.

```
SELECT * FROM ivoa.Obscore
WHERE dataproduct_type='image'
AND s_resolution < 0.3 AND s_ra > 240 AND s_ra < 255
```

```
        AND s_dec > 10 AND s_dec < 11
        AND (em_min > 2.1e-06 AND em_max < 2.4e-06)
            OR(em_min >= 1.6e-06 AND em_max <= 1.8e-06)
            OR(em_min >= 1.2e-06 AND em_max <= 1.4e-06)
```

A similar query could be submitted to a remote TAP service using the *curl* application as follows (in this example a CADC TAP service is referenced):

```
curl -v -L -d "REQUEST=doQuery&LANG=ADQL&QUERY=
select * from ivoa.ObsCore where dataproduct_type='image' AND s_resolution
< .3 AND s_ra >240 and s_ra < 255 AND s_dec > 10 and s_dec < 11
and (em_min > 2.1e-06 AND em_max < 2.4e-06)
OR(em_min >= 1.6e-06 AND em_max <= 1.8e-06)
OR(em_min >= 1.2e-06 AND em_max <= 1.4e-06)"
"http://www.cadc.hia.nrc.gc.ca/caom/sync"
```

A.1   Datasets selection based on self criteria

A.1.1.   Use case 1.1

Any dataset at specified values for energy, position and time duration

Show me all observations satisfying:

     I.   DataType = any
     II. Energy includes 5 keV
     III. RA includes 16.00
     IV. DEC includes +10
     V. Exposure time > 10 ks

```
    SELECT * FROM ivoa.Obscore
    WHERE em_min < 2.48E-10 AND em_max > 2.48 E-10
    AND CONTAINS(POINT('ICRS',16.0,10.0),s_region) = 1
    AND t_exptime > 10000
```

A.1.2.   Use case 1.2

Image or cubes with positions defined in a list, where an X-Ray image and an image in band 500-900 nm can be found.

Let me input a list of RA and DEC coordinates and show me spatially coincident observations satisfying:

     I.   Imaging or spectroscopy data
     II. Includes one or more of the RA,DEC values in an input list (*searchByList* functionality)
     III. Includes both a wavelength in the range 5000-9000 angstroms AND an X-ray image (*IntersectionSet* operation)

This use case may need several steps to select images from X-RAY domain, select image and spectra on optical domain and compute the overlap.

It requires two functionalities from the service:

- *searchByList*, to query on lists of positions given as input
- *InstersectionSet*, to compute the intersection between two response lists.

A.1.3.   Use case 1.3

Images in band J, H, K with spatial resolution > 0.3 arcsec

Show me a list of all observations satisfying:

I. DataType=Image
II. Spatial resolution better than 0.3 arcseconds
III. Filter = J or H or K
IV. RA between 16 hours and 17 hours
V. DEC between 10 degrees and 11 degrees

A.1.4. Use case 1.4

High resolution images in band V, I, Z posed longer than 1000s

Show me a list of all observations that satisfying:

I. DataType=Image
II. Wavelength=V or I or Z
III. Spatial Resolution < 0.7 arcsec FWHM
IV. Exposure Time > 1000 seconds
V. Data Quality: Fully Calibrated

A.1.5. Use case 1.5

IFU for Quasars seen at redshift >3 with radio flux > 1mJy

Show me all data that satisfies:

I. DataType=IFU
II. DataQuality: Fully Calibrated
III. ObjectClass='quasar'
IV. Redshift > 3
V. Radio flux > 1 mJy

We assume here that data providers will expose IFU data using
dataproduct_type='cube'.

```
SELECT * FROM ivoa.Obscore
WHERE dataproduct_type='cube'
AND calib_level > 1
AND CONTAINS(POINT('ICRS', quasar_ra, quasar_dec), s_region) = 1
```

Requires two functionalities from the service: catalog access and searchByList

A.1.6. Use case 1.6

Image data from a list of observations at particular positions and around specified dates.

For an input list of RA, DEC, Modified Julian Date (MJD), show me all data that satisfies

I. DataType=imaging
II. RA,DEC include the value of the list (searchByList) and Observation date is within 1
day of the MJD value

```
SELECT * FROM ivoa.Obscore
WHERE dataproduct_type='image'
AND CONTAINS(POINT('ICRS',user_ra,user_dec), s_region) = 1
AND ABS((t_max + t_min)/2 – user_date ) < 1
```

A.2. Discovering spectra data

A.2.1. Use case 2.1

Spectra with energy bands and above counts threshold

Show me a list of all data that satisfies:

    I.   DataType=Spectrum
    II. Energy spans 1 to 5 keV
    III. Total counts in spectrum > 100
    IV. Exposure time > 10000 seconds
    V. Data Quality: Fully Calibrated

A.2.2.   Use case 2.2

High resolution spectra containing 650 nm with exposure time longer than 1h
Show me a list of all data that satisfies:

    I.   DataType=Spectrum
    II. Wavelength includes 6500 angstroms
    III. Spectral Resolution better than 15 angstroms
    IV. Spatial Resolution better than 2 arcseconds FWHM
    V. Exposure Time > 3600 seconds
    VI. Data Quality = Any

A.2.3.   Use case 2.3

Show me a list of all data that satisfies: Emission line width Halpha > 2000 km/s FWHM
(Catalog search for emission lines and searchByList)

A.3. *Discover multi-dimensional datasets*

A.3.1.   Use case 3.1

High resolution spectral cubes around one position covering more than 100 square arcsec
Show me a list of data with:

    I.   DataType=cube
    II. RA,DEC includes value RA1,DEC1
    III. Field size > 100 square arcseconds
    IV. DataSensitivity = deep
    V. Spectral resolution better than 5 angstroms FWHM

A.3.2.   Use case 3.2

Velocity cubes with resolution better than 50km/s at a defined position
    Show me a list of all data that satisfies:

    I.   DataType=cube with 3 dimensions
    II. Axes includes Velocity
    III. Axes includes RA
    IV. Axes includes DEC
    V. Velocity Resolution better than 50 km/s
    VI. RA includes 16.000
    VII.Dec includes +41.000

    NB: in this case optional data model fields related to redshift axis can be used using
    em_ucd=`spect.DopplerVeloc`, for instance.

A.3.3.   Use case 3.3

Cube including spectral values at 6500 and at 4000 angstroms and resolution better than 5 Angstroms

Show me a list of all data that satisfies:

    I.   DataType=cube
    II.   RA includes 16.00
    III.   Dec includes +41.00
    IV.   Wavelength includes 6500 angstroms
    V.   Wavelength includes 4000 angstroms
    VI.   Spectral resolution better than 5 angstroms
    VII.   Exposure time more than 3600 seconds
    VIII. Data Quality: Fully Calibrated

A.3.4. Use case 3.4

Polarization cubes with sky area more than 100x100 pixels.

Show me a list of all data that satisfies:

    I.   DataType=Cube with 4 dimensions
    II. Axes includes FREQ
    III. Spatial axes contain more than 100 pixels
    IV. Axes includes STOKES
    V. Frequency resolution < 1 MHz
    VI. Rest Frequency = 345.795990 GHz appears in band

A.3.5. Use case 3.5

High spectral resolution with Frequency range more than 500Mhz and sky area more than 100 square pixels

Show me a list of all data that satisfies:

    I.DataType=Cube with 3 dimensions
    II.Axes includes FREQ
    III.Axes includes RA with > 100 pixels
    IV.Axes includes DEC with > 100 pixels
    V.Frequency extent > 500 MHz
    VI.Rest Frequency = 345.795990 GHz appears in band
    VII.The redshift is not specified, but should default to zsource for the target.

NB: I to V are supported in ObsTAP; VI and VII need target redshift properties extracted from catalogs

A.3.6. Use case 3.6

Hyperspectral cubes with more than 200 pixels on each axis and spatial resolution better than 2 arcsec for a list of objects extracted from SDSS data collection.

Show me a list of all data that satisfies:

    I.   DataType=Cube with 3 dimensions
    II. Axes includes WAVE
    III. Axes includes more than 200 pixels along each spatial axis
    I.   Spatial resolution better than 2 arcsec
    II. For a selected list of SDSS objects

A.4. Discovering time series

A.4.1. Use case 4.1

Times series for a sky position, with date, length and exposure constraints

Show me a list of all data which satisfies:

I.   DataType=TimeSeries
II. RA includes 16.00 hours
III. DEC includes +41.00
IV. Time resolution better than 1 minute
V.   Time interval (start of series to end of series) > 1 week
VI. Observation data before June 10, 2008
VII. Observation data after June 10, 2007

A.4.2. Use case 4.2

Times series for a sky position, with a minimum of time slots

This use-case is added in ObsCore 1.1 to search for time series against the number of time slots and be sure to have enough samples along the time axis

Show me a list of all data which satisfies:

I.   DataType=TimeSeries
II. RA includes 16.00 hours
III. DEC includes +41.00
IV. Time resolution better than 1 minute
V.   Time interval (start of series to end of series) > 1 year
VI. Number of time slots > 50

A.4.3. Use case 4.3

Times series for tracking a particular event

Show me a list of all data matching a particular event (gamma ray burst) in time interval and space:

I.   DataType=TimeSeries
II. RA includes 16.00 hours
III. DEC includes +41.00
IV. Time start > MJD 55220. and Time stop < MJD 55221
V.   Number of time slots > 1000

A.5. Discovering event lists

A.5.1. Use case 5.1

Eventlists at position P and time T with exposure time longer than 100000s

Show me a list of all event lists observed for a particular phenomenon with constraints on space and time coverage like:

I.   DataType=event list
II. RA includes target_RA
III. DEC includes target_DEC
IV. Time resolution better than 1 minute
V.   Exposure Time > 100000s
VI. Observation data after June 10, 2007

VII.   Observation data before June 10, 2008

A.5.2.   Use case 5.2

Matching images and event lists from different data collections

Show me a list of all event lists observed in the XMM data collection within the region covered by a particular image of GALEX data collection.

   I.   DataType=event list
   II.   Region contains polygon P1 extracted from the foot print of a GALEX image
   III. Data collection= XMM
   IV.   Instrument name=EPN

A.6. *Discovering general data from collections counterparts*

A.6.1.   Use case 6.1

Optical Imaging around detected objects in M81 group with X-ray and radio image cut-outs

Show me a list of all data that satisfies:

   I.   Optical imaging
   II.   In the M81 group
   III. With area greater than 0.5 square degrees
   IV.   With sensitivity > 10 for point source m=25
   V.   I also want X-ray data with cut-outs 5 arcmin on a side of all the detected galaxies
   VI.   I also want Radio data cut-outs 5 arcmin on a side around detected galaxies

A.6.2.   Use case 6.2

Imaging and spectroscopy around position P in SDSS and CFHTLS data collections

 Show me a list of all data that satisfies:

   I.   DataType=Imaging or Spectroscopy
   II.   RA includes 16.00 hours
   III. DEC includes +41.00 degrees
   IV.   SDSS images and spectra AND CFHTLS images and spectra

A.6.3.   Use case 6.3

Imaging and Xray data for selected objects in Virgo cluster

In Virgo cluster show me imaging and X-ray data for all galaxies that are cluster members and have B < 21.

Strategy: Build up list from catalogs extraction and then compose query as use-case A 1.2

A.6.4.   Use case 6.4

Image, spectra and event list from same data collection around position P and day D

Give me images, spectra and event lists stemming from a common collection around my favorite object, within a range in time.

A.7. *Complex Use Cases*

Here are complex scenarios illustrating some broader perspectives for VO dataset discovery. Such discovery strategies have been sketched for VO demos and training use-cases at VO schools sessions.

A.7.1.   Use case 7.1

Given COSMOS (or other survey) X-Ray source catalog give me all the sources with photoZ > X, and spiral galaxy counterpart and produce radio - to -X-ray SEDs.

Comment: Requires source/object catalogs to drive data query (for SED info which may be catalog or data).

A.7.2.   Use Case 7.2

Given a list of Abell clusters, give me all their Chandra images with exposure time > X, after I select regions occupied by the diffuse emission, give me all the Chandra point sources in these regions, and find their redshift (I want to find background quasars because I am interested in lensing and I have no idea where to go to find z). For the quasars, give me high resolution (< 0.5") optical and radio images, and build SEDs.

Comment: Requires source/object catalogs and interactive image interactions (applications/interfaces), further query, and more catalogs to drive data query.

A.7.3.   Use case 7.3

Find me all the variable Chandra sources with optical counterpart and redshift. If redshift is not available, give me an SED to compare with source templates (I also would like to run a tool or obtain a library of such templates from a theory database, which I expect the VO to provide). My aim is to separate stars from variable quasars.

Comment: Complex use-case, including templates and theory as well as catalogs.

B: ObsCore Data Model Detailed Description

This section provides a full description of all data model elements including both mandatory and optional elements (specified by the value in the "MAN" column). The full UType for all elements of the Observation Core Components data model includes an "*obscore***:**"prefix (defining the namespace for ObsCoreDM) which has been elided here for brevity.

Column Name	Utype	Unit	Type	Description
OBSERVATION INFORMATION (section B.1)				
dataproduct_type	ObsDataset.dataProductType	unitless	enum string	Data product (file content) primary typ
dataproduct_subtype	ObsDataset.dataProductSubtype	unitless	string	Data product specific type
calib_level	ObsDataset.calibLevel	unitless	enum int	Calibration level of th observation: in {0, 1, 3, 4}
TARGET INFORMATION (section B.2)				
target_name	Target.name	unitless	string	Object of interest
target_class	Target.class	unitless	string	Class of the Target object as in SSA
DATA DESCRIPTION (section B.3)				
obs_id	DataID.observationID	unitless	string	Internal ID given by ObsTAP service
obs_title	DataID.title	unitless	string	Brief description of dataset in free forma
obs_collection	DataID.collection	unitless	string	Name of the data collection
obs_creation_date	DataID.date	unitless	date	Date when the datase was created
obs_creator_name	DataID.creator	unitless	string	Name of the creator the data
obs_creator_did	DataID.creatorDID	unitless	string	IVOA dataset identific given by the creator
CURATION INFORMATION (section B.4)				
obs_release_date	Curation.releaseDate	unitless	date	Observation release date (ISO 8601)
obs_publisher_did	Curation.publisherDID	unitless	string	ID for the Dataset gi by the publisher.
publisher_id	Curation.publisherID	unitless	string	IVOA-ID for the Publisher
bib_reference	Curation.reference	unitless	string	Service bibliographic reference
data_rights	Curation.rights	unitless	enum string	Public/Secure/Propri ary/
ACCESS INFORMATION (section B.5)				

access_url	Access.reference	unitless	string	URL used to access dataset
access_format	Access.format	unitless	string	Content format of the dataset
access_estsize	Access.size	Kbyte	Int	Estimated size of dataset: in kilobytes
s_ra	Char.SpatialAxis.Coverage.Location.Coord.Position2D.Value2.C1	Deg	double	Central Spatial Positi in ICRS Right ascension
s_dec	Char.SpatialAxis.Coverage.Location.Coord.Position2D.Value2.C2	Deg	double	Central Spatial Positi in ICRS Declination
s_fov	Char.SpatialAxis.Coverage.Bounds.Extent.diameter	Deg	double	Estimated size of the covered region as the diameter of a contain circle
s_region	Char.SpatialAxis.Coverage.Support.Area	unitless	string	Sky region covered b the data product (expressed in ICRS frame)
s_resolution	Char.SpatialAxis.Resolution.refval.value	Arcsec	double	Spatial resolution of data as FWHM of PSF
s_xel1	Char.SpatialAxis.numBins1	Unitless	integer	Number of elements along the first coordinate of the spa axis
s_xel2	Char.SpatialAxis.numBins2	Unitless	integer	Number of elements along the second coordinate of the spa axis
s_ucd	Char.SpatialAxis.ucd	Unitless	string	UCD for the nature of the spatial axis (pos c u,v data)
s_unit	Char.SpatialAxis.unit	Unitless	string	Unit used for spatial axis
s_resolution_min	Char.SpatialAxis.Resolution.Bounds.Limits.LoLimit	Arcsec	double	Resolution min value spatial axis (FHWM o PSF)
s_resolution_max	Char.SpatialAxis .Resolution.Bounds. Limits.HiLimit	Arcsec	double	Resolution max value spatial axis
s_calib_status	Char.SpatialAxis.calibrationStatus	unitless	Enum string	Type of calibration along the spatial axis
s_stat_error	Char.SpatialAxis.Accuracy.StatError.Refval.value	Arcsec	double	Astrometric precision along the spatial axis
s_pixel_scale	Char.SpatialAxis.Sampling.RefVal.SamplingPeriod	Arcsec	double	Sampling period in world coordinate uni along the spatial axis

t_xel	Char.TimeAxis.numBins	unitless	integer	Number of elements along the time axis
t_min	Char.TimeAxis.Coverage.Bounds.Limits.StartTime	D	double	Start time in MJD
t_max	Char.TimeAxis.Coverage.Bounds.Limits.StopTime	D	double	Stop time in MJD
t_exptime	Char.TimeAxis.Coverage.Support.Extent	S	double	Total exposure time
t_resolution	Char.TimeAxis.Resolution.Refval.valueResolution.Refval.value	S	double	Temporal resolution FWHM
t_calib_status	Char.TimeAxis. calibrationStatus	unitless	Enum string	Type of time coordinate calibration
t_stat_error	Char.TimeAxis.Accuracy.StatError.Refval.value	S	double	Time coord statistical error
SPECTRAL CHARACTERISATION (section B6.2)				
em_xel	Char.SpectralAxis. numBins	unitless	integer	Number of elements along the spectral axis
em_ucd	Char.SpectralAxis.ucd	unitless	string	Nature of the spectral axis
em_unit	Char.SpectralAxis.unit	unitless	string	Units along the spectral axis
em_calib_status	Char.SpectralAxis. calibrationStatus	unitless	Enum string	Type of spectral coord calibration
em_min	Char.SpectralAxis.Coverage.Bounds.Limits.LoLimit	M	double	start in spectral coordinates
em_max	Char.SpectralAxis.Coverage.Bounds.Limits.HiLimit	M	double	stop in spectral coordinates
em_res_power	Char.SpectralAxis.Resolution.ResolPower.refVal	unitless	double	Value of the resolving power along the spectral axis. (R)
em_res_power_min	Char.SpectralAxis.Resolution.ResolPower.LoLimit	unitless	double	Resolving power min value on spectral axis
em_res_power_max	Char.SpectralAxis.Resolution.ResolPower.HiLimit	unitless	double	Resolving power max value on spectral axis
em_resolution	Char.SpectralAxis.Resolution.Refval.value	M	double	Value of Resolution along the spectral axis
em_stat_error	Char.SpectralAxis.Accuracy.StatError.Refval.value	M	double	Spectral coord statistical error
OBSERVABLE AXIS (section B6.4)				
o_ucd	Char.ObservableAxis.ucd	unitless	String	Nature of the observable axis
o_unit	Char.ObservableAxis.unit	unitless	string	Units used for the observable values
o_calib_status	Char.ObservableAxis.calibrationStatus	unitless	Enum string	Type of calibration for the observable coordinate
o_stat_error	Char.ObservableAxis.Accuracy.StatError.Refval.value	units specifie	double	Statistical error on the Observable axis

		d by o_unit		

POLARIZATION CHARACTERISATION (section BB.6.5)				
pol_xel	Char.PolarizationAxis.numBins	unitless	integer	Number of elements along the polarizatio axis
pol_states	Char.PolarizationAxis.stateList	unitless	Enum string	List of polarization states present in the data file

PROVENANCE (section BB.7)				
facility_name	Provenance.ObsConfig.Facility.name	unitless	string	The name of the facili telescope space craft used for the observat
instrument_name	Provenance.ObsConfig.Instrument.name	unitless	string	The name of the instrument used for t observation
proposal_id	Provenance.Proposal.identifier	unitless	string	Identifier of proposal which observation belongs

Table 5: Data model summary (with addition of axes dimension and UCD for each axis)

B.1. Observation Information

This class is a place holder that gathers all metadata relative to an observed and distributed dataset. It points to existing classes of Spectrum DM and types from VODataService (Plante et al. 2010) .

B.1.1. Data Product Type *(dataproduct_type)*

The model defines a data product type attribute for the Observation Class. It is the type of the observation product the user queries for or selects for retrieval. Only high level generic types are defined at this level to allow global data discovery queries to be posed to multiple data archives. This is coded as a string that conveys a general idea of the content and organization of a dataset. Possible values for this string are described in section **3.3.1**. The short name for this attribute is *dataproduct_type* in the implemented Ivoa.ObsCore table, as shown in Table 4.

B.1.2. Data Product Subtype *(dataproduct_subtype)*

In order to be more precise the data product type may be refined with a second field, the data product *subtype*. Unlike the more generic *dataproduct_type*, 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. While less useful for global data discovery this allows the data products within a specific archive to be precisely identified and referenced in queries of that specific archive. The data provider should define a vocabulary sufficient to classify all science data products in their archive to be exposed with ObsTAP. In the future we may be able to define broader standards to classify data at this level, although it will likely always be the case that data differs at the level of specific instrumental survey data collections. The data product subtype allows data within a specific archive or data collection to be precisely classified and referenced in subsequent discovery queries. The list of terms defined for *dataproduct_subtype* labels should be published by the data centers and documented. These can be gathered as an IVOA vocabulary in a RDF dedicated resource for instance.

43

However, in the meantime, the vocabulary used can be discovered easily by a simple query based on 'select distinct …'

B.1.3. Calibration level *(calib_level)*

It is a convention we suggest to use to classify the different possible calibration status of an observed dataset. These 4 categories allow distinguishing 4 levels of calibration and would be sufficient for 80% of the data collections. This will be up to the data provider to consider how to map his own internal classification to the suggested scale here.

See section 3.3.2

B.2. 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 'NULL' if necessary or have a value like 'DARK' for instance to specify a dark field and not a pointed observation.

B.2.1. Target Name (*target_name*)

The *target_name* specified here is often un-reliable when one is searching for specific classes of objects, at least for most archives since it is quite hard to standardize automatically the target name for each observation. So the users could be warned that specifying *target_name* for their search will not necessarily return the expected results. It is quite useful for moving target like planets.

B.2.2. Class of the Target source/object *(target_class)*

This field indicates the type of object that was pointed for this observation. It is a string with possible values defined in a special vocabulary set to be defined: list of object classes (or types) used by the SIMBAD database, NED or defined in another IVOA vocabulary.

B.3. Dataset Description

After acquisition and reduction an observation is uniquely identified by its creator and gets a creator dataset identifier. This information is defined in the Spectrum data model in the *DataID* class. We re-use this class and the Utype DataID.CreatorDID in order to distinguish two datasets 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 the same dataset, with a unique DataID.CreatorDID *obs_creator_did* but possibly different *obs_publisher_did* (given by the data provider). Therefore a unique identifier is needed here. In the ObsCore model, the short name associated to this ID should be *obs_id* (to be checked).

The second identifier used in this model is the one given by the data provider, and defined in the Curation Class.

B.3.1. Creator name (*obs_creator_name*)

The name of the institution or entity which created the dataset, in a simple string.

B.3.2. Observation Identifier *(obs_id)*

The *obs_id* column contains a collection-specific identifier for an observation. In the case where multiple data products are available for an observation (e.g. with different calibration

levels), the *obs_id* value will be the same for each product of the observation. This is equivalent to the dataset name for many archives where dataset name could have many files associated with them.

B.3.3. Dataset Text Description *(obs_title)*

This data model field re-uses a field from the Spectrum Data model: DataID.Title. It should contain a brief description (displayable in less than one line of text) specifying in scientific terms the content of the dataset. The contents of this field are free format and are up to the data provider. For example a radio survey field consisting of HI and CO cubes and an associated 2D continuum image might use *obs_title* to describe the individual data products as "HI cube", "CO cube", "Stokes I continuum image at 1420 MHz", and so forth.

This is commonly used in analysis software to e.g. describe a dataset in a query response table, in a plot header, in the label of a displayed image, and so forth. This helps the user to check the validity and pertinence of a selected dataset for his/her personal goal.

B.3.4. Collection name (*obs_collection*)

The name of the collection (DataID.Collection) identifies the data collection to which the data product belongs. A data collection can be any collection of datasets which are alike in some fashion. Typical data collections might be all the data from a particular telescope, instrument, or survey. The value is either the registered shortname for the data collection, the full registered IVOA identifier for the collection, or a data provider defined *shortname* for the collection. Examples: HST/WFPC2, VLT/FORS2, CHANDRA/ACIS-S, etc.

We understand that this is not a very precisely defined field. What is important for the data provider is to use the collection name which is meaningful to astronomers.

B.3.5. Creation date (*obs_creation_date*)

The date when the dataset was created. This is a time stamp, adopting the VO format regularly used in protocols and defined in DALI (P. D. Dowler 2013) section 3.1.2 , i.e a restriction of the ISO 8601 format without the Z tag , as defined in FITS standard.("YYYY-MM-DDThh:mm:ss").

B.3.6. Creator name (*obs_creator_name*)

The name of the institution or entity which created the dataset.

B.3.7. Dataset Creator Identifier *(obs_creator_did)*

IVOA dataset identifier given by its creator. See definition in the SpectrumDM specification (McDowell, Tody and al 2011)

B.4. Curation metadata

The *Curation* class inherits from the Spectrum data model and VOResource concepts too. The various attributes for ObsCore are:

B.4.1. Publisher Dataset ID *(obs_publisher_did)*

This is the identifier the publisher provides for this observation. It is generally different from the original identifier given by the creator of the dataset. (new reduction, new calibration, etc..). The corresponding Utype mapped from the Spectrum DM is *Curation.PublisherDID* and relates to the same definition.

This field contains the IVOA dataset identifier (Plante and al. 2007) for the published data product. This value must be unique within the namespace controlled by the dataset publisher (data center). It will also be globally unique since each publisher has a unique registered publisher ID. The same dataset may however have more than one publisher dataset identifier if it is published in more than one location (the creator DID, if defined for the given dataset, would be the same regardless of where the data is published).

B.4.2. Publisher Identifier *(publisher_id)*

The IVOA ID for the data provider as defined in the Spectrum DM.

B.4.3. Bibliographic Reference *(bib_reference)*

URL or bibcode for documentation. This is a forward link to major publications which reference the dataset. This is re-used from the SSA definition. See (Tody, Dolensky and al. 2012) in section 4.2.5.6 about Curation Metadata.

B.4.4. Data Rights (*data_rights*)

This parameter allows mentioning the availability of a dataset. Possible values are: *public*, *secure*, or *proprietary* as stated in the VODataService recommendation (Plante et al. 2010).

B.4.5. Release Date (*obs_release_date*)

This is a new attribute added to the original Curation class inherited from the Spectrum Data Model.

It specifies the date of public release for an observation or a data product. This time stamp is a convenient way to distinguish public and private observations and also tell users when a specific data product will become available. The value is in ISO 8601 format reusing this pattern: ("YYYY-MM-DDThh:mm:ss") and could be NULL. An observation with a NULL value in the *releaseDate* attribute is proprietary by definition.

B.5. Data Access

The data format as well as the URL to access the dataset is provided by the **Access** Class inherited from the SSA Utype list (Tody, Dolensky and al. 2012). Also included is an attribute for the estimated size of the data file.

B.5.1. Access Reference *(access_url)*

This item (Access.Reference) contains a URL that can be used to download the data product (as a file of some sort).

Users should be aware that these URL values could be volatile and subject to be different from one access time to another. The access reference URL may refer to a static object or may cause data to be generated on the fly, so long as access is synchronous.

B.5.2. Access Format (access_format)

This data model item is defined in section **4.7** where you can find the list of possible values.

We are aware that many different domains and applications need to define data formats, and then define a controlled vocabulary based on implementation feedback given by data providers at different sites.

B.5.3. Estimated Size (access_estsize)

The Access.Size field contains the approximate size (in kilobytes) of the file available via the *corresponding url.* This is used only to gain some idea of the size of a data product before downloading it, hence only an approximate value is required. It is only a useful indication that can help to tune download functionalities in an application according to high volumes of data and transfer bit rate.

B.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 dataset especially the position, band, time, and the type of observed quantity that we call "observable" in the data model. The **observable axis** can cover various types of flux but also velocity, etc. Such a description was tackled in the IVOA Characterisation data model (Louys and DataModel-WG. 2008) from which we re-use mainly the first and second levels of details except for the spatial coverage where the support region (level 3) is used too.

B.6.1. Spatial axis

B.6.1.1. Spatial sampling: number of elements for each coordinate

The number of spatial bins gives an estimation of the size and richness of a dataset in terms of sample points. This size information about the data will help the user to decide whether to download or extract a subpart from a discovered N-D dataset, like an image or a cube.

If the data portion of a dataset consists of a single array, the actual size of the array should be specified. If the dataset consists of multiple arrays (e.g., a single dataset containing all data elements of a CCD mosaic) then the size of the overall dataset should be specified.

This information is present in the ObsCore DM as the 'numBins' attribute, whose Utype is defined as *Char.SpatialAxis.numBins1 for the first coordinate, and Char.SpatialAxis.numBins2 for the second one.*

B.6.1.2. The observation reference position: *(s_ra and s_dec)*

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.

- Coordinate system

  The coordinate system defined in the Characterisation DM is based on the STC:Coordsys class. The model in principle supports all kind of coordinate systems defined in the STC reference list (Rots 2007). However, the ObsTAP implementation of the model mandates that queries expressed in ICRS should be supported in an ObsTAP service. This allows a general query to be sent to multiple archives or data centers, but requires some interpretation /conversion of coordinates at the server side. Still this is efficient for the large data discovery strategy we need to provide. Data in a specific coordinate system will be available via client applications that would do the conversions or adapt the coordinate system to some specific servers.

- Coordinates

  The model uses the *Location* Class from the Characterisation DM, with the Utype values:

  Char.SpatialAxis.Coverage.Location.Coord.Position2D.Value2.C1

  Char.SpatialAxis.Coverage.Location.Coord.Position2D.Value2.C2

whose short names in the ObsCore table are **s_ra** and **s_dec**. We assume that ObsTAP implements these coordinates in the ICRS system.

Using other coordinate systems as defined in STC (Rots 2007) and re-used in the Characterisation DM can be considered in client applications in charge of the coordinate translations.

B.6.1.3. The covered region

The Coverage class along the spatial axis provides two possible concepts:

- **Bounds** which in turn can use two representations:

    a) **A bounding box** that can estimate very coarsely the coverage of an observation. It is modeled as a couple of intervals on each coordinates with Utypes:

    Char.SpatialAxis.Coverage.Bounds.Limits.LoLimit2Vec.C1

    Char.SpatialAxis.Coverage.Bounds.Limits.HiLimit2Vec.C1

    Char.SpatialAxis.Coverage.Bounds.Limits.LoLimit2Vec.C2

    Char.SpatialAxis.Coverage.Bounds.Limits.HiLimit2Vec.C2

    b) The **extent of the field** of view *(s_fov)*

    The model offers to estimate the size of the diameter of the greater circle encompassing the field of view.

This is not covered by the Characterisation DM v1.1 but in the new release of Characterisation v2.0 as *Char.SpatialAxis.Coverage.Bounds.Extent.diameter*, a new definition added in Characterisation DM v2.0 (to appear).

- **Support**: (s_region)

    A precise region description of spatial footprint of the dataset using region types like Circle, Polygon, etc., provided in STC. The Utypes:

    Char.SpatialAxis.Coverage.Support.Area
    Char.SpatialAxis.Coverage.Support.AreaType

    define this region.

    Depending on the version of the TAP service used for distributing the data, this DM element will be mapped differently: see details in the TAP specifications, version 1.0 and above)

B.6.1.4. Spatial Resolution (*s_resolution*)

The minimal size that can be distinguished along the spatial axis, *s_resolution* is specified in arcseconds and has the following Utype: *Char.SpatialAxis.Resolution.Refval.value*

When this value is difficult to evaluate or inadequate, the range of possible resolution can be given in the optional data model fields: *s_resolution_min* and *s_resolution_max, as shown in Table 7.*

B.6.1.5. Astrometric Calibration Status: (*s_calib_status*)

A string to encode the calibration status along the spatial axis (astrometry).

Possible values could be **{uncalibrated, raw, calibrated}** and correspond to the Utype *Char.SpatialAxis.calibrationStatus*

For some observations, only the pointing position is provided (*s_calib_status* ="uncalibrated"). Some other may have a raw linear relationship between the pixel coordinates and the world coordinates (*s_calib_status* ="raw").

B.6.1.6. Astrometric precision (*s_stat_error*)

This parameter gives an estimate of the astrometric statistical error after the astrometric calibration phase. The corresponding Utype is: *Char.SpatialAxis.Accuracy.StatError.Refval.value*

B.6.1.7. Spatial sampling (*s_pixel_scale*)

This corresponds to the sampling precision of the data along the spatial axis. It is stored as a real number corresponding to the spatial sampling period, i.e., the distance in world coordinates system units between two pixel centers. It may contain two values if the pixels are rectangular.

B.6.2. 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, *Char.SpectralAxis.ucd* named as **em_ucd** in ObsTAP optional fields. Possible values for this UCD are defined in the Spectrum DM (McDowell, Tody and al 2011) in section 4.1.

Depending on the UCD used to specify the axis, the ObsCore model allows to describe the spectral coordinates in a relevant unit, corresponding to the spectral quantity, and specified in the model in *Char.SpectralAxis.unit* **(em_unit)**

Here is a short list of preferred value for the Observation data model Core Components extracted from the recommended values proposed in the Spectrum DM.

Spectral coordinate	Char.SpectralAxis.ucd	Char.SpectralAxis.unit
Frequency	em.freq	Hz
Wavelength	em.wl	m or angstrom
Energy	em.energy	keV, J, erg

Note that for the ObsTAP implementation, the Spectral axis coordinates are constrained as a wavelength quantity expressed in meters as mentioned in section 4.17

B.6.2.1. Number of spectral sampling elements *(em_xel)*

Number of values spanned along the spectral axis, corresponding to Utype

Char.SpectralAxis.numBins

B.6.2.2. Spectral calibration status (*em_calib_status*)

This attribute of the spectral axis indicates the status of the data in terms of spectral calibration. Possible values are defined in the Characterisation Data Model and belong to **{uncalibrated , calibrated, relative, absolute}.**

B.6.2.3. Spectral Bounds

These are the limits of the spectral interval covered by the observation, in short *em_min* and *em_max.*

These limiting values are compatible with definitions of the physical quantity defined in the ucd and unit fields.

In the ObsTAP implementation such values are expressed as wavelength but using meters as units, as it is easily convertible.

B.6.2.4. Spectral Resolution

As in the Characterisation data model we distinguish a reference value of the point spread function along the spectral axis from the resolution power along this axis, more appropriate when the resolution varies along the spectral axis. Only one of the following is needed in the data model:

a) A reference value for **Spectral Resolution (*em_resolution*)**

A mean estimate of the resolution, e.g. Full Width at Half Maximum (FWHM) of the Line Spread Function (or LSF). This can be used for narrow range spectra whereas in the majority of cases, the resolution power is preferable due to the LSF variation along the spectral axis. The corresponding Utype is *Char.SpectralAxis.Resolution.Refval.value.*

b) A reference value for **Resolving Power (*em_res_power*)**

This is an average estimation for the spectral resolution power stored as a double value, with no unit.   *Char.SpectralAxis.Resolution.ResolPower refval*

c) **Resolving Power** limits *(em_res_power_min, em_res_power_max)*

These parameters simply give the limits of variation of the resolution power in the observation as minimal and maximal values and use the following Utypes:

Char.SpectralAxis.Resolution.ResolPower.LoLimit

Char.SpectralAxis.Resolution.ResolPower.HiLimit

B.6.2.5. Accuracy along the spectral axis *(em_stat_error)*

This is also provided in the Characterisation data model, using the item mapped to the Utype: *Char.SpectralAxis.Accuracy.StatError.Refval.value* and, stored in the same units as all the other spectral quantities.

B.6.2.6. Doppler/Redshift datasets

Dataset including an axis representing Doppler velocity (e.g., a velocity cube) can be discovered using this specification. This can be indicated by specifying an appropriate value for the optional *em_ucd* attribute, defining the type of velocity on the axis. The following UCD values are defined to represent velocities:

Ucd	Definition
spect.dopplerVeloc.opt	*Radial velocity derived from a wavelength shift using the optical convention*
spect.dopplerVeloc.radio	*Radial velocity derived from a frequency shift using the radio convention*

spect.dopplerVeloc.rel	Radial velocity, derived using the relativistic convention
spect.doppler.z	Redshift derived from a spectral feature

If **em_ucd** contains one of these strings, then the discovered datasets can be any velocity or redshift data product. However the spectral coverage of the dataset should still be specified in *em_min, em_max* as the vacuum wavelength in meters, as detailed in section B.6.2.3. Obscore v1.1 does not support the full description of properties along a redshift axis but allows to discover data sets expressed in velocity units or optical redshift but characterized by a coverage in wavelength, typically a wavelength interval around one particular emission line.

B.6.3. Time axis

B.6.3.1. Time coverage *(t_min, t_max, t_exptime)*

Three time stamps are used: *t_min*, *t_max*, usually equals to start and stop time and *t_exptime* the exposure time. These **must** be stored in MJD format, much preferable for easy calculations. Other information is given in subsection   4.14 and 4.15.

B.6.3.2. Time resolution (*t_resolution*)

Estimated or average value of the temporal resolution with Utype

> *Char.TimeAxis.Resolution.Refval.value*

B.6.3.3. Time axis: number of sampling elements (*t_xel*)

Number of values spanned along the time axis, corresponding to Utype

Char.TimeAxis.numBins. For a single shot observation, an image or spectrum or cube , *t_xel* equals 1, and for a time series, t_xel contains the number of different shots, and can be a valuable criterium to retain or discard a dataset for further time varying analysis.

B.6.3.4. Time Calibration Status: (*t_calib_status*)

This parameter gives the status of time axis calibration. This is especially useful for time series.

Possible values are principally **{ uncalibrated, calibrated, raw, relative}.** This may be extended for specific time domain collections.

B.6.3.5. Time Calibration Error: (*t_stat_error*)

A parameter used if we can estimate a statistical error on the time measurements (for time series again). This value is expressed in seconds.

B.6.4. Observable Axis:

B.6.4.1. Nature of the observed quantity (*o_ucd*)

Most observations measure some flux quantity depending on position, spectral coordinate, 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 specified by the UCD attached to this axis.

The possible UCD values are part of the UCD1+ vocabulary (Preite Martinez, et al. 2007) . One can find simple flux classes like: *phot.flux, phot.flux.density, phot.count, phot.mag,* or more complex combinations such as: *phot.flux.density;phys.polarization.stokes*

Various possibilities have been gathered at the following URL:

http://www.ivoa.net/internal/IVOA/ObsTap/ListForObservable25Oct2010.pdf

which provides a (non-exhaustive) list of possible triplets (observable name, UCD, units) for various observables data providers may want to describe in their archive. The units used to encode values of the Observable quantity are specified in *Char.ObservableAxis.unit* and can be exposed in ObsTAP with the optional field **o_unit**. See examples of unit strings in the table mentioned above.

B.6.4.2. Calibration status on observable (Flux or other) *(o_calib_status)*

This describes the calibration applied on the Flux observed (or other observable quantity).

It is a string to be selected in **{absolute, relative, normalized, any}** as defined in the SSA specification (Tody, Dolensky and al. 2012) in section 4.1.2.10.

This list can be extended or updated for instance using an extension mechanism similar to the definition of new UCDs in the IVOA process, following the feedback from implementations of ObsTAP services.

B.6.5. Polarization measurements (*pol_states, pol_xel*)

This covers the case when the observed flux was recorded for various states of a polarizer. Then the dataset can be: a set of images, a set of spectra, a set of spectral cubes with various polarization flux at each data point. What differs is the nature of the observable and the list of possible polarization states recorded.

In this case, *o_ucd* should at least contain the substring '**phys.polarisation**' like in "*phot.flux.density;phys.polarisation*". The polarization measure can be specified with UCD strings like in "*phot.flux.density;phys.polarisation.Stokes.I*", etc. as shown in the list of observable UCD cited above.

B.6.5.1. List of polarization states *(pol_states)*

In order to gather the polarization information, we define a polarization axis which is degenerated as compared to other axes, but describes necessary polarization properties of the dataset. *Char.PolarizationAxis.stateList* contains the list of the various polarization modes present in the dataset.

In the Obs/TAP implementation the column name is **pol_states**.It is a mandatory field with NULL value allowed if no polarization applies. Otherwise it contains a list of polarization labels inspired from the FITS specification. See Table 7 in FITS WCS Paper 1 (Greisen et Calabretta 2002) . Labels are combined using symbols from the {I Q U V RR LL RL LR XX YY XY YX POLI POLA} set and separated by a **/** character. A leading **/** character must start the list and a trailing / character must end it. It should be ordered following the above list, compatible with the FITS list table for polarization definition.

Then a query can be easily written like:

```
SELECT * WHERE pol_states LIKE '%Y%'
```

which brings back all polarization moments of type :Y XY YX YY

On the contrary,

```
SELECT * WHERE pol_states LIKE '%/Y/%'
```

selects only datasets containing Y polarization state.

See A. Richards IVOA Note for the context of polarization data (Richards et Bonnarel 2010).

B.6.5.2. Number of polarization elements *(pol_xel)*

pol_xel specifies the number of different polarization states present in the data. Its Utype in the Obscore DM is *Char.PolarizationAxis.numBins.* The default value is 0, indicating that polarization was not explicitly observed.

If an effective polarization measurement has been made, then *pol_xel* should be specified as 1 or greater and *pol_states* should contain the list of states recorded in the dataset.

B.6.6. Additional Parameters on Observable axis

When implementing an ObsTAP service, the archive manager may need to publish some parameters not present in the current version of ObsCore.

As an example, the type of noise present in an observation is not modeled. It depends on the instrument, on the data collection and can be defined in an optional column name in the IVOA.Obscore table like:

Column Name	Datatype	Size	Units	ObsCoreDM Utype	UCD	Princ.	Index	Std
o_stat_error_type	adql:VARCHAR	20	NULL	NULL	stat.error;meta. code	*1*	*0*	*0*

Possible values of o_stat_error_type could be: {poisson, gauss, speckle} and mentioned in the description of additional columns (See section 4.22 for more details)

o_ stat_error _mean, o_ stat_error _sigma can be defined as the parameters for the Gaussian case

o_ stat_error_poisson can be defined as the Poisson gain, etc.

In case of these optional fields, defined by the data provider, the Utype column in the ObsCore table has a NULL value.

B.7. Provenance

Provenance contains a class to represent the entire Observing configuration used to acquire an observation. Instrumental parameters are gathered here.

B.7.1. Facility (*facility_name*)

The Facility class codes information about the observatory or facility used to collect the data. In this model we define one attribute of Utype *Provenance.ObsConfig.facility.name* which re-uses the Facility concept defined in the VODataService specification (Plante et al. 2010).

 For combined observations stemming from multiple facilities the name may contain a list of comma separated strings, or the word "Many"; if the list is too long, as defined in the VODataservice specification.

The definition of a list of possible name values could be a task for the IVOA Semantic working group, starting from various facilities and instruments lists resources published in the community.

B.7.2. Instrument name (*instrument_name*)

The name of the instrument used for the acquisition of the observation. It is given in the model *as Provenance.ObsConfig.instrument.name* and encoded as a string. The possible name values could be checked in coordination with the Semantic WG too. Multiple values are also allowed for complex observations as defined for **facility name**.

B.7.3. Proposal (*proposal_id*)

Each proposal has an identifier attribute that can be used to collect all observations and data products related to the same proposal. The corresponding Utype will simply be *Provenance.Proposal.identifier .*

[NB: Here is presented only a minimal set of information on the instrumental configuration. See future documents on Provenance data model.]

Appendix C: TAP_SCHEMA tables and usage

C.1 Implementation Examples

There are various implementations of ObsCore in TAP, following the evolution of the TAP protocol and ADQL.

CADC and XMM have reference implementations for ObsCore 1.0 available via the VO application TapHandle 2.0 (http://saada.unistra.fr/taphandle/) for instance.

These sites are currently updating their ObsTAP service to version 1.1.

Check the registry for ObsTAP services serving ObsCore v 1.1 metadata

C.1.1.   ObsCore 1.0 first examples

Examples of the ObsTAP use-cases and ObsTAP Schema can be found at the following URL:

   http://www.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/cvo/ObsCore/

C.1.2.   Implementing a package of multiple data products

This example shows how to describe a complex observation dataset, referenced by its *obs_id* field and containing different data products, all packed together in an archive file.

For example, for High Energy datasets we could have as the table response of an ObsTAP query:

obs_id	data product type	data product subtype	Calibration Level	Access Format	Title
123	event	chandra.hrc.pkg	1	application/x-tar-gzip	Chandra ACS-XYZ observation package (event,refimage)
123	image	chandra.hrc.refimage	2	image/fits	ACS-XYZ reference image
123	image	chandra.hrc.preview	2	image/jpeg	Chandra ACS-XYZ preview image
345	event	rosat.foo.pkg	1	application/x-tar-gzip	ROSAT observation package

The subtype could in principle be more generic but will likely be instrument-specific for a level 1 data product.

The Title should concisely describe the data product, e.g., origin, instrument, ID, what it is (observation package, calibration, standard view, etc.). The title string is what one normally wants to output on a displayed image or plot to identify to a human the data being shown. Its length is limited to one line of text.

C.2.   List of data model fields in TAP_SCHEMA

TAP Schema (`TAP_SCHEMA.columns`) metadata for all mandatory and optional data model fields are given in the following tables. We suggest using only lower case for all column names in the tables used to implement ObsTAP, in order to simplify queries against multiple database systems.

Utypes strings are easier to read in 'Camel case' that is why we recommend using these strings as written in the tables below, for all interactions with users and developers. These strings are produced and read in VOTable for instance and may be consumed by some applications.

For Utypes originating from the Spectrum Data model, we keep the original writing.

For Utypes created from the UML ObsCore model, we apply these rules:

- Attributes of a class start with a lower case letter (e.g. calibrationStatus)
- For classes referencing one other class, we use the name of the reference or role, and not the one of the pointed class.

The meaning of the various columns corresponds to the definitions of the TAP IVOA standard (Dowler, Tody et Rixon, Table Access Protocol 2010). See section 2.6.3 for the description of columns attributes.

As a reminder, the last three columns are implementation oriented:

'principal': means that this item is of main importance, and for instance is recommended in a select or should be shown in first priority in a query response.

This is not equivalent to the mandatory status of the field. All mandatory fields are principal, but some optional fields may be considered 'principal'.

'std': means this column is defined by some IVOA standard as opposed to a customized metadata defined by a specific service.

'indexed': tells if this column can be used as table index to optimize queries. Possible values for each of these three fields are integers, with this convention: (0=false, 1=true).

Table 6 `TAP_SCHEMA.columns` values for the mandatory fields of an ObsTAP table. All Utypes have the data model namespace prefix **"obscore:"** omitted in the table. The Datatype, Size, Principal, Index, and Std values shown here are informative for TAP 1.0 only; later versions of TAP may specify different values.

Column Name	Datatype	Size	Units	ObsCoreDM Utype	UCD	Principal	Index	Std
dataproduct_type	adql:VARCHAR	TBD	NULL	ObsDataset.dataProductType	**meta.code.class**	1	TBD	1
calib_level	adql:INTEGER	NULL	NULL	ObsDataset.calibLevel	meta.code;obs.calib	1	TBD	1
obs_collection	adql:VARCHAR	TBD	NULL	DataID.collection	meta.id	1	TBD	1
obs_id	adql:VARCHAR	TBD	NULL	DataID.observationID	meta.id	1	TBD	1
obs_publisher_did	adql:VARCHAR	TBD	NULL	Curation.publisherDID	**meta.ref.ivoid**	1	TBD	1
access_url	adql:CLOB	NULL	NULL	Access.reference	meta.ref.url	1	0	1
access_format	adql:VARCHAR	NULL	NULL	Access.format	meta.code.mime	1	0	1
access_estsize	adql:BIGINT	NULL	kbyte	Access.size	phys.size;meta.file	1	0	1
target_name	adql:VARCHAR	TBD	NULL	Target.name	meta.id;src	1	0	1
s_ra	adql:DOUBLE	NULL	deg	Char.SpatialAxis.Coverage.Location.Coord.Position 2D.Value2.C1	pos.eq.ra	1	0	1
s_dec	adql:DOUBLE	NULL	deg	Char.SpatialAxis.Coverage.Location.Coord.Position 2D.Value2.C2	pos.eq.dec	1	0	1
s_fov	adql:DOUBLE	NULL	deg	Char.SpatialAxis.Coverage.Bounds.Extent.diameter	phys.angSize;instr.fov	1	0	1
s_region	adql:REGION	NULL		Char.SpatialAxis.Coverage.Support.Area	pos.outline;obs.field	1	0	1
s_resolution	adql:DOUBLE	NULL	arcsec	Char.SpatialAxis.Resolution.Refval.value	pos.angResolution	1	TBD	1
s_xel1	adql:BIGINT	NULL	NULL	Char.SpatialAxis.numBins1	meta.number	1	TBD	1
s_xel2	adql:BIGINT	NULL	NULL	Char.SpatialAxis.numBins2	meta.number	1	TBD	1
t_min	adql:DOUBLE	NULL	d	Char.TimeAxis.Coverage.Bounds.Limits.StartTime	time.start;obs.exposure	1	0	1
t_max	adql:DOUBLE	NULL	d	Char.TimeAxis.Coverage.Bounds.Limits.StopTime	time.end;obs.exposure	1	0	1
t_exptime	adql:DOUBLE	NULL	s	Char.TimeAxis.Coverage.Support.Extent	time.duration;obs.exposure	1	TBD	1
t_resolution	adql:DOUBLE	NULL	s	Char.TimeAxis.Resolution.Refval.value	time.resolution	1	0	1
t_xel	adql: BIGINT	NULL	NULL	Char.TimeAxis.numBins	meta.number	1	TBD	1

em_min	adql:DOUBLE	NULL	m	Char.SpectralAxis.Coverage.Bounds.Limits.LoLimit	em.wl;stat.min	1	0	1
em_max	adql:DOUBLE	NULL	m	Char.SpectralAxis.Coverage.Bounds.Limits.HiLimit	em.wl;stat.max	1	0	1
em_res_power	adql:DOUBLE	NULL	NULL	Char.SpectralAxis.Resolution.ResolPower.refVal	spect.resolution	1	TBD	1
em_xel	adql: BIGINT	NULL	NULL	Char.SpectralAxis.numBins	meta.number	1	TBD	1
o_ucd	adql:VARCHAR	TBD	NULL	Char.ObservableAxis.ucd	meta.ucd	1	0	1
pol_states	adql:VARCHAR	TBD	NULL	Char.PolarizationAxis.stateList	meta.code;phys.polarization	1	0	1
pol_xel	adql: BIGINT	NULL	NULL	Char.PolarizationAxis.numBins	meta.number	1	TBD	1
facility_name	adql:VARCHAR	NULL	NULL	Provenance.ObsConfig.Facility.name	meta.id;instr.tel	1	TBD	1
Instrument_name	adql:VARCHAR	NULL	NULL	Provenance.ObsConfig.Instrument.name	meta.id;instr	1	TBD	1

Column Name	Datatype	Size	Units	ObsCoreDM Utype	UCD	Principal	Index	Std
dataproduct_subtype	adql:VARCHAR	TBD	NULL	ObsDataset.dataProductSubtype	**meta.code.class**	1	TBD	1
target_class	adql:VARCHAR	TBD	NULL	Target.class	src.class	1	TBD	1
obs_creation_date	adql:TIMESTAMP	NULL	NULL	DataID.date	time;meta.dataset	1	TBD	1
obs_creator_name	adql:VARCHAR	TBD	NULL	DataID.creator	meta.id	1	TBD	1
obs_creator_did	adql:VARCHAR	TBD	NULL	DataID.creatorDID	meta.id	0	TBD	1
obs_title	adql:VARCHAR	200	NULL	DataID.title	meta.title;obs	1	0	1
publisher_id	adql:VARCHAR	TBD	NULL	Curation.publisherID	**meta.ref.ivoid**	1	TBD	1
bib_reference	adql:VARCHAR	TBD	NULL	Curation.reference	meta.bib	0	0	1
data_rights	adql:VARCHAR	NULL	NULL	Curation.rights	meta.code	0	0	1
obs_release_date	adql:TIMESTAMP	NULL	NULL	Curation.releaseDate	time.release	1	0	1
s_ucd	adql:VARCHAR	NULL	NULL	Char.SpatialAxis.ucd	meta.ucd	1	0	1
s_unit	adql:VARCHAR	NULL	NULL	Char.SpatialAxis.unit	meta.unit	1	0	1
s_resolution_min	adql:DOUBLE	NULL	arcsec	Char.SpatialAxis.Resolution.Bounds.Limits.LoLimit	pos.angResolution;stat.min	1	0	1
s_resolution_max	adql:DOUBLE	NULL	arcsec	Char.SpatialAxis.Resolution.Bounds.Limits.HiLimit	pos.angResolution;stat.max	1	0	1
s_pixel_scale	adql:DOUBLE	NULL	arcsec	Char.SpatialAxis.Sampling.RefVal.SamplingPeriod	phys.angSize;instr.pixel	1	0	1
s_calib_status	adql:VARCHAR	NULL	NULL	Char.SpatialAxis.calibrationStatus	meta.code.qual	1	0	1
s_stat_error	adql:DOUBLE	NULL	arcsec	Char.SpatialAxis.Accuracy.StatError.Refval.value	stat.error;pos.eq	0	0	1
t_calib_status	adql:VARCHAR	NULL	NULL	Char.TimeAxis.calibrationStatus	meta.code.qual	0	0	1
t_stat_error	adql:DOUBLE	NULL	s	Char.TimeAxis.Accuracy.StatError.Refval.value	stat.error;time	0	0	1
em_ucd	adql:VARCHAR	NULL	NULL	Char.SpectralAxis.ucd	meta.ucd	1	TBD	1
em_unit	adql:VARCHAR	NULL	NULL	Char.SpectralAxis.unit	meta.unit	1	0	1
em_calib_status	adql:VARCHAR	NULL	NULL	Char.SpectralAxis.calibrationStatus	meta.code.qual	0	0	1
em_res_power_min	adql:DOUBLE	NULL	NULL	Char.SpectralAxis.Resolution.ResolPower.LoLimit	spect.resolution;stat.min	1	0	1

em_res_power_max	adql:DOUBLE	NULL	NULL	Char.SpectralAxis.Resolution.ResolPower.HiLimit	spect.resolution;stat.max	1	0	1
em_resolution	adql:DOUBLE	NULL	m	Char.SpectralAxis.Resolution.Refval.value	spect.resolution;stat.mean	1	0	1
em_stat_error	adql:DOUBLE	NULL	m	Char.SpectralAxis.Accuracy.StatError.Refval.value	stat.error;em	0	0	1
o_unit	adql:VARCHAR	NULL	NULL	Char.ObservableAxis.unit	meta.unit	1	0	1
o_calib_status	adql:VARCHAR	NULL	NULL	Char.ObservableAxis.calibrationStatus	meta.code.qual	1	TBD	1
o_stat_error	adql:DOUBLE	NULL	o_unit	Char.ObservableAxis.Accuracy.StatError.Refval.value	**stat.error**	0	0	1
proposal_id	adql:VARCHAR	NULL	NULL	Provenance.Proposal.identifier	meta.id; obs.proposal	0	TBD	1

Table 7 `TAP.schema.columns` values for the optional fields for an ObsTAP table. All Utypes have the data model namespace prefix **"obscore:"** omitted in the table.

C.3 Examples of ObsTAP query responses

Here is a Query Response VOTable document obtained by the CADC ObsTAP/TAP1.1 service at

http://www.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/sia/v2query?MAXREC=0

It contains all mandatory fields for ObsCore1.1. and the implementation details for the Datalink service attached to the TAP response.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<VOTABLE version="1.2" xmlns="http://www.ivoa.net/xml/VOTable/v1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ivoa.net/xml/VOTable/v1.2
http://www.ivoa.net/xml/VOTable/VOTable-1.2.xsd">
<RESOURCE type="results">
<INFO name="QUERY_STATUS" value="OK" />
<INFO name ="QUERY" value="SELECT * FROM ivoa.ObsCore WHERE dataproduct_type IN ('image', 'cube')" />
<TABLE>
<FIELD name="dataproduct_type" datatype="char" ucd="meta.code.class" utype="obscore:ObsDataset.dataProductType" xtype="adql:VARCHAR" arraysize="128*">
    <DESCRIPTION>type of product</DESCRIPTION>
</FIELD>
<FIELD name="calib_level" datatype="int" ucd="meta.code;obs.calib" utype="obscore:ObsDataset.calibLevel" xtype="adql:INTEGER">
  <DESCRIPTION> calibration level (0,1,2,3,4)</DESCRIPTION>
</FIELD>
<FIELD name="obs_collection" datatype="char" ucd="meta.id" utype="obscore:DataID.collection" xtype="adql:VARCHAR" arraysize="128*">
  <DESCRIPTION>short name for the data colection</DESCRIPTION>
</FIELD>
<FIELD name="facility_name" datatype="char" ucd="meta.id;instr.tel" utype="obscore:Provenance.ObsConfig.Facility.name" xtype="adql:VARCHAR" arraysize="128*">
  <DESCRIPTION>telescope name</DESCRIPTION>
</FIELD>
<FIELD name="instrument_name" datatype="char" ucd="meta.id;instr" utype="obscore:Provenance.ObsConfig.Instrument.name" xtype="adql:VARCHAR" arraysize="128*">
  <DESCRIPTION>instrument name</DESCRIPTION>
</FIELD>
<FIELD name="obs_id" datatype="char" ucd="meta.id" utype="obscore:DataID.observationID" xtype="adql:VARCHAR" arraysize="128*">
  <DESCRIPTION>internal dataset identifier</DESCRIPTION>
</FIELD>
<FIELD name="obs_release_date" datatype="char" ucd="time.release" utype="obscore:Curation.releaseDate" xtype="adql:TIMESTAMP" arraysize="*">
    <DESCRIPTION>timestamp of date the data becomes publicly available</DESCRIPTION>
</FIELD>
<FIELD name="access_url" datatype="char" ucd="meta.ref.url" utype="obscore:Access.reference" xtype="adql:CLOB" arraysize="*">
    <DESCRIPTION>URL to download the data</DESCRIPTION>
</FIELD>
<FIELD name="access_format" datatype="char" ucd="meta.id;class" utype="obscore:Access.format" xtype="adql:VARCHAR" arraysize="128*">
```

```xml
    <DESCRIPTION> format of the data file(s) </DESCRIPTION> </FIELD>
<FIELD name="access_estsize" datatype="int" ucd="phys.size;meta.file" unit="kB" utype="obscore:Access.size" xtype="adql:INTEGER">
  <DESCRIPTION>estimated size of the download</DESCRIPTION>
</FIELD>
<FIELD name="target_name" datatype="char" ucd="meta.id;src"       utype="obscore:Target.name" xtype="adql:VARCHAR" arraysize="32*">
  <DESCRIPTION>name of intended target</DESCRIPTION>
</FIELD>
<FIELD name="s_ra" datatype="double" ucd="pos.eq.ra" unit="deg" utype="obscore:Char.SpatialAxis.Coverage.Location.Coord.Position2D.Value2.C1" xtype="adql:DOUBLE">
  <DESCRIPTION>RA of central coordinates</DESCRIPTION>
</FIELD>
<FIELD name="s_dec" datatype="double" ucd="pos.eq.dec" unit="deg" utype="obscore:Char.SpatialAxis.Coverage.Location.Coord.Position2D.Value2.C2" xtype="adql:DOUBLE">
  <DESCRIPTION>DEC of central coordinates</DESCRIPTION>
</FIELD>
<FIELD name="s_fov" datatype="double" ucd="phys.angSize;instr.fov" unit="deg" utype="obscore:Char.SpatialAxis.Coverage.Bounds.Extent.diameter" xtype="adql:DOUBLE">
     <DESCRIPTION>size of the region covered (~diameter of minimum bounding circle)</DESCRIPTION>
</FIELD>
<FIELD name="s_region" datatype="char" ucd="pos.outline;obs.field" unit="deg" utype="obscore:Char.SpatialAxis.Coverage.Support.Area" xtype="adql:REGION" arraysize="*">
  <DESCRIPTION>region bounded by observation</DESCRIPTION>
</FIELD>
<FIELD name="s_resolution" datatype="double" ucd="pos.angResolution" unit="arcsec" utype="obscore:Char.SpatialAxis.Resolution.Refval.value" xtype="adql:REAL">
     <DESCRIPTION>typical spatial resolution</DESCRIPTION>
</FIELD>
<FIELD name="s_xel1" datatype="long" ucd="meta.number" utype="obscore:Char.SpatialAxis.numBins1" xtype="adql:BIGINT">
  <DESCRIPTION>dimensions (number of pixels) along one spatial         axis</DESCRIPTION>
</FIELD>
<FIELD name="s_xel2" datatype="long" ucd="meta.number" utype="obscore:Char.SpatialAxis.numBins2" xtype="adql:BIGINT">
  <DESCRIPTION>dimensions (number of pixels) along the other spatial   axis</DESCRIPTION>
</FIELD>
<FIELD name="t_min" datatype="double" ucd="time.start;obs.exposure" unit="d" utype="obscore:Char.TimeAxis.Coverage.Bounds.Limits.StartTime"  xtype="adql:DOUBLE">
  <DESCRIPTION>start time of observation (MJD)</DESCRIPTION>
</FIELD>
<FIELD name="t_max" datatype="double" ucd="time.stop;obs.exposure" unit="d" utype="obscore:Char.TimeAxis.Coverage.Bounds.Limits.StopTime" xtype="adql:DOUBLE">
  <DESCRIPTION>end time of observation (MJD)</DESCRIPTION>
</FIELD>
<FIELD name="t_exptime" datatype="double" ucd="time.duration;obs.exposure" unit="s" utype="obscore:Char.TimeAxis.Coverage.Support.Extent" xtype="adql:DOUBLE">
  <DESCRIPTION>exposure time of observation</DESCRIPTION>
</FIELD>
<FIELD name="t_resolution" datatype="double" ucd="time.resolution" unit="s" utype="obscore:Char.TimeAxis.Resolution.Refval.value" xtype="adql:DOUBLE">
  <DESCRIPTION>typical temporal resolution</DESCRIPTION>
</FIELD>
<FIELD name="t_xel" datatype="long" ucd="meta.number" utype="obscore:Char.TimeAxis.numBins" xtype="adql:BIGINT">
  <DESCRIPTION>dimensions (number of pixels) along the time axis</DESCRIPTION>
</FIELD>
```

```xml
<FIELD name="em_min" datatype="double" ucd="em.wl;stat.min" unit="m" utype="obscore:Char.SpectralAxis.Coverage.Bounds.Limits.LoLimit" xtype="adql:DOUBLE">
  <DESCRIPTION>start spectral coordinate value</DESCRIPTION>
</FIELD>
<FIELD name="em_max" datatype="double" ucd="em.wl;stat.max" unit="m" utype="obscore:Char.SpectralAxis.Coverage.Bounds.Limits.HiLimit" xtype="adql:DOUBLE">
  <DESCRIPTION>stop spectral coordinate value</DESCRIPTION>
</FIELD>
<FIELD name="em_res_power" datatype="double" ucd="spec.resolution" utype="obscore:Char.SpectralAxis.Resolution.ResolPower.refVal" xtype="adql:DOUBLE">
    <DESCRIPTION>typical spectral resolution</DESCRIPTION>
</FIELD>
<FIELD name="em_xel" datatype="long" ucd="meta.number" utype="obscore:Char.SpectralAxis.numBins" xtype="adql:BIGINT">
  <DESCRIPTION>dimensions (number of pixels) along the energy        axis</DESCRIPTION>
</FIELD>
<FIELD name="em_ucd" datatype="char" ucd="meta.ucd" utype="obscore:Char.SpectralAxis.ucd" xtype="adql:VARCHAR" arraysize="32*">
  <DESCRIPTION>UCD describing the spectral axis</DESCRIPTION>
</FIELD>
<FIELD name="pol_states" datatype="char" ucd="phys.polarization" utype="obscore:Char.PolarizationAxis.stateList" xtype="adql:VARCHAR" arraysize="32*">
  <DESCRIPTION>polarization states present in the data</DESCRIPTION>
</FIELD>
<FIELD name="pol_xel" datatype="long" ucd="meta.number" utype="obscore:Char.PolarizationAxis.numBins" xtype="adql:BIGINT">
  <DESCRIPTION>dimensions (number of pixels) along the polarization axis</DESCRIPTION>
</FIELD>
<FIELD name="o_ucd" datatype="char" ucd="meta.ucd" utype="obscore:Char.ObservableAxis.ucd" xtype="adql:VARCHAR" arraysize="32*">
  <DESCRIPTION>UCD describing the observable axis (pixel      values)</DESCRIPTION>
</FIELD>
<FIELD name="core_id" datatype="long" xtype="adql:BIGINT">
  <DESCRIPTION>primary key</DESCRIPTION>
</FIELD>
<FIELD name="lastModified" datatype="char" xtype="adql:TIMESTAMP" arraysize="*">
  <DESCRIPTION>time stamp of last modification of the metadata</DESCRIPTION>
</FIELD>
<FIELD name="obs_publisher_did" datatype="char" ID="caomPlaneURI" ucd="meta.ref.ivoid" utype="obscore:Curation.publisherDID" xtype="adql:VARCHAR" arraysize="256*">
  <DESCRIPTION>publisher dataset identifier</DESCRIPTION>
</FIELD>
<DATA>
 <TABLEDATA />
</DATA>
</TABLE>
  <INFO name="QUERY_STATUS" value="OK" />
</RESOURCE>
<RESOURCE type="meta" utype="adhoc:service">
 <PARAM name="resourceIdentifier" datatype="char" arraysize="*" value="ivo://cadc.nrc.ca/datalink" />
 <PARAM name="standardID" datatype="char" arraysize="*" value="ivo://ivoa.net/std/DataLink#links-1.0" />
 <PARAM name="accessURL" datatype="char" arraysize="61" value="http://www.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/caom2ops/datalink"    />
```

```xml
<GROUP name="inputParams">
 <PARAM name="ID" datatype="char" ref="caomPlaneURI" arraysize="*" value="" />
 <PARAM name="RESPONSEFORMAT" datatype="char" arraysize="*"  value="application/x-votable+xml;content=datalink">
  <VALUES>
   <OPTION value="application/x-votable+xml;content=datalink" />
   <OPTION value="application/x-download-manifest+txt" />
  </VALUES>
 </PARAM>
</GROUP>
</RESOURCE>
</VOTABLE>
```