International
Virtual
Observatory
Alliance

IVOA Cube-Data Access Layer Implementation note

Version 1.0

IVOA Note 2016-02-19

Working group
DAL
This version
http://www.ivoa.net/documents/Cube-DAL/20160219
Latest version
http://www.ivoa.net/documents/Cube-DAL
Previous versions

Author(s)
Markus Demleitner, François Bonnarel
Editor(s)

Abstract

This document describes how several DAL protocol can be implemented in combination or separately for discovery, description and access of multi-dimensional data. These functionalities can be achieved using ObsTAP ((Louys et al., 2011)), SIAV2 ((Dowler, Tody and Bonnarel, 2015)), DataLink ((Dowler, Bonnarel, Michel and Demleitner, 2015)) and SODA ((Bonnarel et al., 2015)).

Status of This Document

This is an IVOA Note expressing suggestions from and opinions of the authors. It is intended to share best practices, possible approaches, or other perspectives on interoperability with the Virtual Observatory. It should not be referenced or otherwise interpreted as a standard specification.

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

Contents

Acknowledgments

The authors would like to thank all the participants in DAL- WG discussions for their ideas, critical reviews, and contributions to this document.

1 Introduction

Nowadays DAL landscape is very modular. We have created or are developping a pretty large number of different protocols and specifications during the thirteen last years and the most recent ones work in complementarity on some definite aspects and also have to collaborate for higher order and more complex fonctionalities.

DALI ((Dowler et al., 2013)), for example, is a generic specification gavering a set of common definitions across protocols.

But the main differenciation in the DAL landscape comes with TAP emerging beside the so called "Simple Access protocols". TAP and ADQL coupled with the simplified observation datamodel ObscOre allow to develop the standard generic dataset discovery type of services: the so-called "ObsTAP" ((Louys et al., 2011)).

SIAV1.0 ((Tody and Plante, 2009)) and SSA ((Tody et al., 2012)) have two functionalities : data discovery and some hints of data access (cuting-out, resampling, regridding, mosaicking)

On another side the idea of linking discovered or known datasets to additional resources has been in the air since a long time. Obviously there is a need to give access to additional information, other data formats, additional processing. In SIAV1.0 and SSA it could be done by duplicating the lines dedicated to a specific dataset in the Query response. But it was obvioulsly not an efficient and elegant solution. The concept of DataLinking is born from that need.

For data cubes and multi-dimensional data, and later for time series or event list, the Simple protocols were insuficient for search and selection on some axes. On the other side ObsTAP services are not designed at all for data access and only allow full retrieval.

DataLink ((Dowler, Bonnarel, Michel and Demleitner, 2015)) was the right way to link datasets discovered by ObsTAP to various resources and specifically to an access data service -currently SODA ((Bonnarel et al., 2015))- able to provide all kind of cutout and selection facilities (and more access functionalities later).

Simple access protocols were long time providing both discovery and accessData methods with a similar interface. Discovery with a simple HTTP interface still have advantages over ObsTAp for some use cases and some data providers and was decided at some point to develop SIAV2.0 ((Dowler, Tody and Bonnarel, 2015)) version for that.

The current architecture is the following : ObsTAP services and SIAV2 services are parallel discovery (and description) pathes for cube datasets The query response is based on Obscore datamodel in both cases. Full metadata Resource going beyond ObsCore will come later in SIAV2.1 (the idea is it should be a serialization of CubeDM ((Tody et al., 2015))). AccessData (now called SODA) is independant and accessible from the query responses through one of the DataLink mechanisms. The first version is oriented towards Cuting-out and Selection on the axes. More sophisticated access data operations will come later.

Let's see how it works.

2 Data Discovery

This section clarifies the various ways a dataset can be discovered. For implementers choice of developping an SIAV2 or ObsTAP service depends from many factors. Although many providers have implemented both (SIAV2 being a layer built on top of an ObstAP service,allowing some sort of "PQL" queries) it is not required to implement an OBsTAP service to implement an SIAV2 service. For example it could be rather easy to upgrade an old SIAV1 service towards and SIAV2.0 one.

2.1 Discovery via an SIAV2 service

SIAV2 services ((Dowler, Tody and Bonnarel, 2015)) are queriable by setting contraints on the four "axes" of the data (2D-space, spectral, time and polarization) and their properties. Other archive or data features are also possible (target, collection, facilities, etc...). The successfull response is a VOTABLE containing a mandatory result TABLE and optional service descriptor resource(s). The result table describes the dataset characterization on the different axes and give additional operative features. the various columns are mapped from the OBscore 1.1 data model ((Louys et al., 2016)). The accessReference and AccessFormat fields give different solutions to reach the dataset for download or access. This can be done in two ways:

- The data discovery step has yielded a result with the media type

  `application/x-votable+xml;content=datalink`.

  To the client, this indicates that what is given in the access reference (e.g., the `access_url` column in ObsTAP or SIA version 2) is a datalink document (see below for DataLink functionalities).

- IN case the media type is not

  `application/x-votable+xml;content=datalink`

  the access reference is a direct link to the dataset download. To enable additional Datalink functionalities, DAL services can add a serivce descriptor in the DAL response that indicates the availability of a Datalink *service* accompanying the DAL service, looking more or less like this:

```
<RESOURCE type="results">
  [a result from services like TAP, SIA, SSA]
  <TABLE>
    [in particular, we have one field like]
    <FIELD ID="primaryID" name="pubDID" datatype="char" arraysize="*">
      <DESCRIPTION>The publisher DID for the dataset</DESCRIPTION>
    </FIELD>
    ...
  </TABLE>
</RESOURCE>
<RESOURCE type="meta" utype="adhoc:service">
  <PARAM name="standardID" datatype="char" arraysize="*"
    value="ivo://ivoa.net/std/DataLink#links-1.0" />
  <PARAM name="accessURL" datatype="char" arraysize="*"
    value="http://example.com/mylinks/get" />
```

```
  <GROUP name="inputParams">
    <PARAM name="ID" datatype="char" arraysize="*"
      value="" ref="primaryID"/>
  </GROUP>
</RESOURCE>
```

It references one (or more) field(s) from the DAL response.[1] This is explained in more detail in section 4.2 of the DataLink recommendation 1.0. The net result is that DataLink-enabled clients can find ancillary data and use SODA services for data access by virtue of being able to retrieve Datalink documents.

For implementers of SIAV2 services (and ObstAP services −see below) the choice between one of the two strategies exposed above depends on various factors such as :

- size of the dataset: full download of huge datasets may become optional
- dataset type: direct download may be preferred for 2D images, while it can be postponed in the DataLink response for cubes.
- potential number and relative importance of links with respect to the dataset itself.

2.2 Discovery via an ObsTAP service

An ObsTAP service is a TAP service exposing the ivoa.obscore table (ref). The full table schema is the same as the one retrieved by a query on a SIAV2 service. Of course the query is expressed with an ADQL command and the result may actually contain any subset of columns selected in the whole table. In order to get sufficient information for further download and access it is recommanded to retrieve format and reference columns, as well as the main "caracterisation" ones (s_ra, s_dec, s_region, em_min, em_max, t_min, t_max, pol_states) because they may be needed for reuse in further acccess data operations.

The path to full image download and to various links is made via one of the two mechanisms described above for SIAV2. The access reference to the DataLink service mechanism is straightforward. The service descriptor embedded in the OBstAP response is allowed because ObsTAP is a peculiar TAP service and TAP allows the TAP response to contain additional resources in addition to the "results" one (See section 2.9 of TAP1.0 spec). Hence the inclusion of a service descriptor service is valid.

[1] That pattern can be used within the DataLink document as in sect. 4.1, too, to refer to Datalink's ID column, which lets services use a constant access URL in the SODA descriptor.

2.3 other miscelaneous discovery methods

Dataset can also be discovered in various other ways. The "discovery" concept is defined as the discovery of the ivoa publisher_did of a dataset or by the a priori knowledge of a DataLink service and an obsid specific to this DataLink service. Beside the main discovery path via ObsTAP or SIAV2, the following scenarii are possible:

- The discovery can be done via an SIA1.0 or a SSA service. A DataLink service descriptor should have been added to the Service query response to allow further guiding to additional resources
- The discovery can be done via a paper in a journal implementing publication of datasets associated with articles. The editor should describe or implement a DataLink service for further usage.
- Logs of observations are dataset descriptions belonging to the catalog category. They can be exposed in the VO using Cone Search or TAP. They are in general not consistent with the ObsCore model. However they allow some kind of "dataset discovery".
- Hips is an emerging IVOA standard for a global and hierarchical allsky access to pixel and catalogue data ((Fernique et al., 2015)). HipS cells contain ids for original dataset which have been processed to build the Healpix maps. This can be a new way to discover datasets and start a DataLink session to find out additional associated resources.

3 DataLink: links resource

The links web service capability allows to retrieve ressources associated to a dataset (or a set of datasets) the identifier of which are given as value to the standard ID parameter. The response is a VOTABLE describing the resources ((Dowler, Bonnarel, Michel and Demleitner, 2015)). Basically we can associate the dataset with fixed links (eg : another format, some additional metadata, a progenitor, subparts of the datasets, calibration data such as flat, dark or bias fields, etc...) and dynamical links such as custom and standard services. SODA ((Bonnarel et al., 2015)) is a typical standard service which can be provided for a dataset.

An attractive implementation strategy for small-to-medium sized installations is to pre-generate the datalink files. In that way, no extra endpoint is required besides the discovery service and the SODA service.

For fixed links the URL to the associated resource is directly given by the reference column

For dynamical links the links response contains a ref id value in the service_def column. This points to a resource looking like this

```
<RESOURCE type="meta" utype="adhoc:service" ID="TheService">

  [<PARAM name="standardID" datatype="char" arraysize="*"
       value="ivo://ivoa.net/std/TheService#sync-1.0" /> This PARAM only ther if we have a standard servic


  <PARAM name="accessURL" datatype="char" arraysize="*"
       value="http://example.com/my-svcs/theserv/sync?ID=ivo://example.com/data?ds1" />

  <GROUP name="inputParams">
     <PARAM name="par1" ucd="...." datatype="..."
          arraysize="*" />
     <PARAM name="par2" ucd="...." unit="..." datatype="..."
          arraysize="*"/>
     <PARAM name="par3" ucd="....."
          unit="..;" datatype="double"
          arraysize="*" xtype="...." />
     <PARAM name="par4" ucd="...." datatype="char"
          arraysize="*" />
  </GROUP>
</RESOURCE>
```

Of course, the service is free to choose the VOTable ID of the resource with the utype `adhoc:service`;

From the Datalink row with `#proc` semantics, the client sees that there is a service for the dataset in question (identified here through its publisher DID, `ivo://example.com/data?ds1`).

In case we are facing a custom service the "standardID param is absent. Alternatively when we are facing a standard service the inputPARAMS GROUP may be OPTIONAL because these parameters are fully described in the specification of the service and also in its VOSI description available from the registry. However providing these descriptions may allow dynamical recognition by clever clients and will be really usefull in the case standard parameters are not mandatory. The service descriptor will actually only declare the parameters it actually supports.

The case of the SODA service is detailed below.

4 Accessing data: SODA Operation

SODA services may be discovered through Registry queries. However, usually client discover them in Datalink (Dowler, Bonnarel, Michel and Demleitner, 2015) declarations, which can either be standalone or embedded within other services' responses.

Since this pattern can appear somewhat confusing at first, this introductory (non-normative) chapter discusses the usage scenarii for SODA services.

In parallel, we provide advice on the server-side implications of these scenarios.

In all cases, the first step is dataset discovery; when used below, this term could refer to, for instance, SIA, SSA, or ObsTAP, but also to some sort of resolution engine for persistent identifiers (see section 2).

4.1 Datalink only discovery

In this scenario, what is given in the access reference is a datalink document (see section 2.1). Within that document, there is a SODA service descriptor written as specified by Datalink. The whole document would look somewhat like this:

of course, this needs descriptions and ranges; if this text is accepted for the main standard, MD will fill this in.

```
<RESOURCE type="results">
  [datalink links, one of them being]
  <TR>[id=ivo://example.com/data?ds1 service-def=soda; semantics=#proc]</TR>
</RESOURCE>

<RESOURCE type="meta" utype="adhoc:service" ID="soda">

  <PARAM name="standardID" datatype="char" arraysize="*"
       value="ivo://ivoa.net/std/SODA#sync-1.0" />

  <PARAM name="accessURL" datatype="char" arraysize="*"
       value="http://example.com/my-svcs/soda/sync?ID=ivo://example.com/data?ds1" />

  <GROUP name="inputParams">
     <PARAM name="POS" ucd="phys.angArea;obs" datatype="char"
          arraysize="*" />
     <PARAM name="BAND" ucd="em.wl" unit="m" datatype="double"
          arraysize="*"/>
     <PARAM name="TIME" ucd="time.interval;obs.exposure"
          unit="d" datatype="double"
          arraysize="*" xtype="interval" />
     <PARAM name="POL" ucd="meta.code;phys.polarization" datatype="char"
          arraysize="*" />
  </GROUP>
</RESOURCE>
```

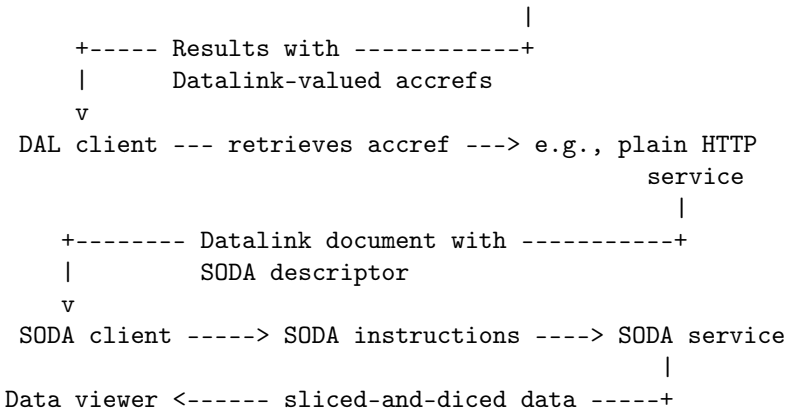
From the service descriptor's standardID *PARAM* it learns that the service's parameters follow the rules laid down here, in particular as regards the three-factor semantics. For instance, the client is guaranteed that BAND, with UCD `em.wl` and unit meters actually denotes the parameter controlling where a cutout on the spectral axis will happen.

SODA's role here is exactly this guarantee of a specific semantics, as opposed to a non-standard service that could use BAND in an entirely different way.

If people think this is a good idea, I'll do SVGs of these

Here is a sketch of the query pattern in this case:

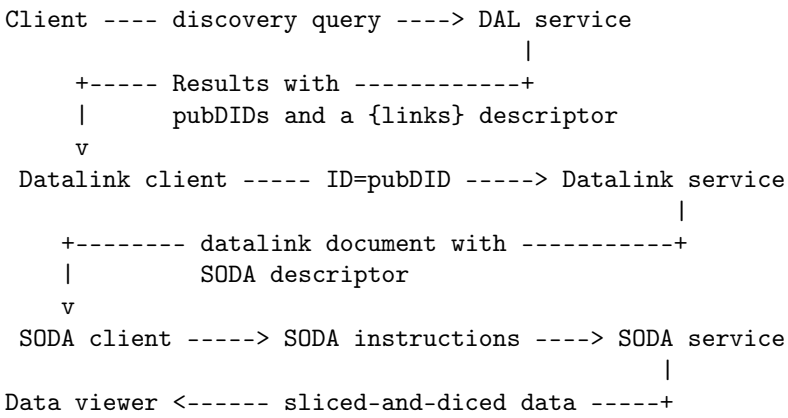
```
Client ---- discovery query ----> DAL service
```



4.2 Datalink Discovery parallel to full access

In many situations services should deliver conventional (e.g., FITS) data products to clients via the access reference field (for example legacy clients would ignore DataLink content and would not be able to find their way to dataset retrieval). To enable SODA and other Datalink functionalities, DAL services can add a service descriptor in the DAL response as it has been explained in 2.1

The query pattern in this case looks like this:



4.3 SODA description in Query response

In some situations, the extra request to retrieve the datalink document for each dataset is not necessary because no other "links" are provided than the SODA one, while the client may have sufficient information to operate the SODA service based on common metadata.

In such cases a service may provide a shortcut by including a SODA descriptor directly in the DAL response. In essence the resulting descriptor

looks like a union of the one given in sect. 4.1 and the one given in sect. 4.2: It includes the SODA parameters, the ID parameter with the reference to the column to take the publisher DID from, but it has a SODA standardID from sect. 4.1 rather than the Datalink one from sect. 4.2.

Sidestepping the extra datalink request might appear attractive in principle, the difficulty of determining the useful parameter ranges make this pattern only interesting in relatively few special cases. Clients must not rely on the presence of full SODA descriptors in DAL responses. Normal SODA operation follows the pattern given in sects. 4.1 and 4.2.

The query pattern here is:

```
Client ---- discovery query ----> DAL service
                                      |
     +----- Results with ------------+
     |      SODA descriptor
     v
 SODA client -----> SODA instructions ----> SODA service
                                                |
Data viewer <------ sliced-and-diced data -----+
```

4.4 SODA autodescription

SODA can autodescribe itself and its parameters. This will be necessary for example, when the service will be accessed coming from a discovery via the registry. In that case it will provide the service descriptor (see above for definition) when it is queried without Query parameters or with the unique ID parameter.

In the latter case SODA PARAMETER description and domain will be related to the dataset metadata as designed in Obscore. (to be defined if an agreement can be found).

A Changes from Previous Versions

A.1 Changes from WD-AccessData-1.0-20140312

This is the initial document version created reusing an explanotary subsection made by Markus Demleitnzer for SODA specification.

References

Bonnarel, F., Demleitner, M., Dowler, P. and Tody, D. (2015), 'Simple image access specification', IVOA Working draft.
URL: *http://www.ivoa.net/documents/SODA/20151221/WD-SODA-1.0-20151221.pdf*

Dowler, P., Bonnarel, F., Michel, L. and Demleitner, M. (2015), 'Datalink', IVOA recommendation.
URL: *http://www.ivoa.net/documents/DataLink/*

Dowler, P., Demleitner, M., Taylor, M. and Tody, D. (2013), 'Data access layer interface, version 1.0', IVOA Recommendation.
URL: *http://www.ivoa.net/documents/DALI*

Dowler, P., Tody, D. and Bonnarel, F. (2015), 'Simple image access specification', IVOA Recommendation.
URL: *http://www.ivoa.net/documents/latest/SIA.html*

Fernique, P., Allen, M., Boch, T., Durand, D., Michel, L., Salgado, J. and Stoehr, F. (2015), 'Hips – hierarchical progressive survey, version 1.0', IVOA Note.
URL: *http://www.ivoa.net/documents/Notes/HiPS/index.html*

Louys, M., Bonnarel, F., Schade, D., Dowler, P., Micol, A., Durand, D., Tody, D., Michel, L., Salgado, J., Chilingarian, I., Rino, B., de Dios Santander, J. and Skoda, P. (2011), 'Observation data model core components and its implementation in the Table Access Protocol, version 1.0', IVOA Recommendation.
URL: *http://www.ivoa.net/documents/ObsCore/20111028/REC-ObsCore-v1.0-20111028.pdf*

Louys, M., Tody, D., Dowler, P., Durand, D., Michel, L., Bonnarel, F., Micol, A. and the IVOA DataModel working group (2016), 'Observation data model core components and its implementation in the Table Access Protocol, version 1.0', IVOA Working draft.
URL: *http://www.ivoa.net/documents/ObsCore/*

Tody, D., Bonnarel, F., Laurino, O., Louys, M., Rots, A., Ruiz, J. E., Selgado, J. and the IVOA Data Model Working Group. (2015), 'Ivoa n-dimensional cube model, version 1.0', IVOA Working draft.
URL: *http://www.ivoa.net/documents/NDimCubeDM/20150320/index.html*

Tody, D., Dolensky, M., McDowell, J., Bonnarel, F., Budavari, T., Busko, I., Micol, A., Osuna, P., Salgado, J., Skoda, P., Thompson, R. and Valdes, F. (2012), 'Simple spectral access protocol version 1.1', IVOA Recommendation.
URL: *http://www.ivoa.net/documents/SSA/20120210/REC-SSA-1.1-20120210.htm*

Tody, D. and Plante, R. (2009), 'Simple image access specification', IVOA Recommendation.
URL: *http://www.ivoa.net/documents/SIA/20091116/*