

# Datalink recognition outside Obscore context



---

F.Bonnarel

on behalf of M.Louys,G.Landais,Pierre Fernique



# The datalink issue (excerpt of Chaitra presentation)

The screenshot shows the ALADIN interface with a table of DataLink records. The table has columns for 'access url', 'sense def', 'error message', 'samantics', 'description', 'content type', 'content length', and 'reusable'. The records are listed in a way that suggests they are not being recognized as DataLink objects.

access url	sense def	error message	samantics	description	content type	content length	reusable
http://www.eso.org			#input		application/ovits	1898385	true
http://www.eso.org			#output		application/ovits		

The field with the Datalink url is not recognized as such. Client cannot prepare appropriate behavior (DataLink popup window)

DataLink table is displayed in a wrong mode / not recognized as such

The screenshot shows the ALADIN interface with a table of DataLink records. The table has columns for 'id', 'access url', 'sense def', 'error message', 'samantics', 'description', 'content type', 'content length', and 'reusable'. The records are listed in a way that suggests they are being recognized as DataLink objects.

id	access url	sense def	error message	samantics	description	content type	content length	reusable
1	http://www.eso.org			#input		application/ovits	1898385	true
2	http://www.eso.org			#output		application/ovits		

# Possible solutions

## 1) Asterics Hackathon one

- Add a complement to the url ucd of the field

```
<FIELD name="url" ucd="meta.ref.url;meta.type.datalink" datatype="char" arraysize="128*">
```

- Pros :

- Self consistent

- Cons :

- Ucd approach Difficult to generalize.  
Fuzzy+fuzzy not always accurate meaning
- Doesn't work if url format changes from line to line



# Possible solutions

## 2) additional obscure access utype

- Add a « access.datalink » utype to ObsCore
  - Pros :
    - self consistent
    - Let ucd usage free
  - Cons :
    - Modification of ObsCore
    - Doesn't work if url format changes from line to line



# Possible solutions

## 3) VOTABLE solution : LINK

- use the LINK element specifying a FIELD:
  - `<FIELD...> <LINK content-type="xxx" href="xxx" ...>`
  - Example for a FITS image:
    - `<FIELD name="Image" ucd="meta.ref.url" datatype="char" arraysize="1">`
    - `<DESCRIPTION>[YN] Epic image of this observation (FITS)</DESCRIPTION>`
    - `<LINK content-type="image/fits" title="Image" href="http://vizier.u-strasbg.fr/viz-bin/nph-htx/A?%5cvizContent%7b$(Image)foo&bar"/>`
    - `</FIELD>`
    - ...
  - Current list of content-types : image/fits, spectrum/fits, catalog/fits , etc..
  - Behavior of application (Aladin) changes according to that
  - Add content-type = "votable/xml;datalink"
- Pros :
  - Extends already working functionality.
  - URL templating
  - Self consistent
- Cons :
  - Doesn't work if url format changes from line to line





# Possible solutions

## 4) ObsCore-like solution

- Add a format column + other ObsCore-like columns

```
--
--
-- <RESOURCE ID="yCat_102009" name="B/xmm">
--   <DESCRIPTION>XMM-Newton Observation Log (XMM-Newton Science Operation Center, 2012)</DESCRIPTION>
--   <CODSYS ID="J2000" system="Eq_FKS" equinox="J2000"/>
--   <TABLE ID="B_xmm_xmmlog" name="B/xmm/xmmlog">
--     <DESCRIPTION>The XMM-Newton Observation log (2017-04-24) vizContent{image/fits}</DESCRIPTION>
--     --
--     <GROUP utype="Obs.obsdataset.accessblock">
--       <FIELDREF ref="A" utype="Obs.obsdataset.data.producttype" />
--       <FIELDREF ref="B" utype="Obs.obsdataset.access.format" />
--     <FIELDREF ref="C" utype="Obs.obsdataset.access.reference" />
--     <FIELDREF ref="D" utype="Obs.dataID.title" />
--   </GROUP>
--   --
--   <FIELD ID="A" name="Product">
--     <DESCRIPTION>Product type (image | spectre | timeseries | document)</DESCRIPTION>
--   </FIELD>
--   <FIELD ID="B" name="Format">
--     <DESCRIPTION>Encoding format</DESCRIPTION>
--   </FIELD>
--   <FIELD ID="C" name="Image" ucd="meta.ref.uri" datatype="char" arraysize="1">
--     <DESCRIPTION>[YN] Epic image of this observation (FITS)</DESCRIPTION>
--   </FIELD>
--   <FIELD ID="D" name="Label">
--     <DESCRIPTION> Product label </DESCRIPTION>
--   </FIELD>
--   --
--   <DATA><TABLEDATA>
--     <TR>
--       --
--       <TD>image</TD>
--       <TD>fits</TD>
--       <TD>http://vizier.u-strasbg.fr/viz-bin/nph-htx?myimage3145</TD>
--       <TD>image 3145
```

- Pros :
  - Utype/ucd complementarity
  - Allows variability from line to line
  - Full description
- Cons :
  - Requires adding fields (or params) and a group



# Possible Evolution of dataset discovery and access



---

F.Bonnarel (CDS)



# Current multidimensional data protocols (SIAV2 set)

- A full family of bounded protocols
  - SIA2.0
  - ObsTAP with Obscore 1.1
  - DataLink
  - SODA
- Main properties
  - ObsTAP and SIA2 allow archived dataset discovery
    - Constraints on all four data axes (spatial, spectral, time and pol) by ADQL (ObsTAP/Core) or PQL (SIA2.0)
  - SODA only allows cutouts and selection
  - Glue among those is made by DataLink technology
  - DALI compliant / sync and async / ucd 1+ and utypes + xtypes





# Quick look to older protocols

## SIAS1 and SSA

- SCS not relevant for datasets
- SIAS1 had no standard possibility to query on BAND / TIME / POL
- SIAS1 had old style ucd, no utypes, no DALI compliancy, etc..
- SIAS1 had virtual data discovery functionality including rebinning/reprojection on provided WCS → not in SIAS2 set
- SSA also provided some virtual data discovery (apparently insufficient / see Petr Talk) ---> not in SIAS2 set
- SSA had specific spectral input parameters eg VARAMP and REDSHIFT ---> not in SIAS2 set
- SSA had more output fields (target description , accuracy on all axes) than ObsCore.
- --> protocols less achieved but with a few things still missing in « SIAS2 set »



# TimeSeries requirements

Don't forget : top 1 CSP priority

- For Discovery (see my talk in TDIG/DAL/DM session)
  - Basic Obscore +
    - Time standard deviation
    - Time sampling location, bounds, standard deviation
    - Time frequency characterisation
    - Variablity, period ???
    - Target name and class
  - Virtual data discovery : TimeSeries has to be created from the database content by the query



# TimeSeries requirements

Don't forget : top 1 CSP priority

- For access (see also ASTERICS requirements on Thursday)
  - Delivering set of ND points with generally sparsed time axis and one to several dependant axes (flux, velocity, position, ... spectra, images ....)
  - Provenance of ND point or ND point additional metadata
  - Time scale / time frame description.
  - MJD representation
  - periodograms



# Spectra (see Petr talk)

- More input parameters for targets
- Virtual data discovery
  - to avoid 2 step discovery and SODA access via DataLink and get similar discovery/access parameters
- Extension of standard SODA to spectra
- More functionalities in SODA (formats, rebinning, axis transformation)
- Etc...



# Images and cubes

- Virtual image (or cube) discovery ( à la HEASARC « Skyview »)
- Pixel cutouts and rebinning/reprojection
- HiPS  $\leftrightarrow$  « SLAV2 set of services » back and forth combination





# Towards a new Discovery protocol ?

- Instead of SIA2.1 + SSA2 +TS1.0, new DsDisc protocol defined by
  - ObsCore extensions
    - Extension of Obscore for TS metadata
    - Extension of Obscore for spectra
    - Extension for others : visibilities ,
  - New dataproduct type specific PQL parameters
  - Virtual data discovery = access.reference is a « best match » SODA Url



# SODA 1.1 ? /SODA 2 ?

- Valid for TimeSeries, spectra, others
- Providing rebinning/reprojection and pixel cutout
- HiPS combination
- Providing extended metadata (special dataset view)
  - using datamodel (VO-DML) serialization
  - Used to be in SIAV2.1 getGoryDetails → getMetadata
- Forced by requested dataset representation
  - ObsCore / Dataset/Cube DM

