IVOA Nov 2023 DAL 2

Time: Sat 11 Nov 16:00 MST **[session #8]**

Active attendees: Markus Demleitner (MD), Dave Morris (DM), Trey Roby (TR), Gilles Landais (GL), Pierre Le Sidaner (PlS), Grégory Mantelet (GM), Mark Taylor (MT), Joshua Fraustro (JF), Marco Molinaro (MM), James Dempsey (JD)

New User Defined Functions for ADQL

- Operator specific prefix for operator functions (e.g. gavo_)
- ivo_ for endorsed functions
- version 1.1 - adds ivo_epoch_prop_pos and ivo_histogram
- Version 1.2 (already in DaCHS and ESAC with gavo prefix)
    - ivo_normal_random
    - ivo_simbadpoint
        - note cannot be a column ref to avoid excessive load on simbad
    - ivo_to_jd, ivo_to_mjd - convert db timestsamp to jd/mjd
    - ivo_transform
        - Convert POINT, CIRCLE, POLYGON between reference frames (e.g. Galactic to ICRS, see refframe vocab)
    - ivo_epoch_prop - 6-vector propogation of position, motion
- See PR on https://github.com/ivoa-std/udf-catalogue
- Call for UDFs to promote to ivo_

RH: ivo_normal_random - don't see a seed, so can't repeat a sequence
MD: In a relational DB, result order is not defined so couldn't reproduce a sequence even with a seed
RH: Transforming geometries - what's the model?
MD: All transforms are simple rotations - note FK4 simple - full FK4 would be problematic. Use case is not high precision but for finding things.

DM: Can you call ivo_transform to ivo_geometry_transform to allow future transforms
MD: Yes, please make the change on github

TR: ivo_simbadpoint - can you support multiple systems?
GL: Yes, we probably have to enlarge this to support NED, IMCCE, epn. Could add wider lookup function
MD: Takes it away from easy. Could have look up table. Might be best to have different functions per resolver

PlS: How would someone know of functions being avaialble?
MD: Can be seen in topcat and PyVO - declared in capabilities endpoint of TAP and tapregext

JS: How is ivo_simbadpoint implemented?
MD: In the translation layer. Database doesn't know about this, hence why can't be a col ref

GM: Plan to add some timestamp and array support in ADQL 1.2

DAP/SODA status report

- New WD for SIA published on github and mailing list since May interop - to help discussion
- SIA:
    - Query by MOC
        - take MOC result and query SIA with it
    - Retieve data in standard mode or in full mode - ask for cutouts directly - can be scripted
    - Rename to DAP - has been discussed before (SIA 2.1, SSA 2.0)
        - DPTYPE allows time series discovery and access
    - Make FACILITY, COLLECTION, INSTRUMENT (only!) case sensitive
    - Suggest keyword to allow wildcards for ivoa_id
    - Root URL query will provide list of supported DPTYPES, FACILITY, INSTRUMENT, COLLECTION
- SODA
    - PIXELS:
        - Specifiy cutout in pixel coords (e.g. repeat cutout on multiopel images after getting world cords from first)
    - RESPONSEFORMAT
        - Allow format conversion e.g. for visualisation, or for extract of votable
        - Could also ask for FITS from HiPS (ala Hips2Fits)
    - DPTYPE
        - Services already allow different DPTYPE to request spectra from cube
        - Moving to standardise this
    - Transformation
        - Change resolution (spectral or spatial)
    - METADATA
        - e.g. extract FITS header, obscore record

In each case one big pull request being built up

Hierarchical DALI examples in TOPCAT and pyVO

- Sales pitch!

- What to do when too many examples in your TAP service
- Note: Examples are really popular and useful with astronomers - encouragement to write some for your service
- Miller's law - 7+-2 items in a mennu
- Continuation allows having a sub menu - to group examples and reduce menu size
    - <a href=".." property="continuation"
- Supported by topcat and pyvo
- Thinking about improvements to presentation in pyvo

GM: Can you nest/cascade menus?
MD: Up to 7 levels in pyvo
MT: Yes topcat nests

XW: Are these in the TAP service or in regtap?
MD: In the examples endpoint of the TAP service

DALI-1.2 WD status

- xtypes
    - optional attribute to indicate a structured type
    - e.g. fields/column in VOTABLE
    - interval can support array of intervals (arraysize="2x*")
    - New ones
        - uri
        - uuid
        - hms
        - dms
        - range
        - shape - polymorphic:circle, range, polygon
        - moc: MOC-2.0 ASCII serialisation
        - multipolygon - union of non overlapping polygons,
    - Support growing for new xtypes
        - CADC/CANFar
        - Vizier
        - TOPCAT
    - Used in SIA-2.0 and SODA-1.0
    - Obscore-next could make use in place of adql:REGION

DM: With the MOC - is this a trap? Pinning generic name to specific version and serialisation
MD: Totally relaxed about that - don't see serialisation changing and already has wide support. Could add moc64 if a base64 serialisation happens
DM: Sure, but take moc 2.0 out of definition
MD: Need version for reference, later version would be moc3
PD: Would need new xtype if serialisation changes anyway

MM: Should we consider the json format - much more powerful than ascii in database
MD: In db, serialisation is binary so this is more about VOTABLE
DM: Not just this standard, across all, if a specific version is required, it should have it in the name to help newcomers work out which to use
JF: if it is written moc the user will immediately refer to the last version of the standard. This is particularly a problem when you work with an old table built with an older version of MOC (and having a different serialization).

TR: xtype have circle and shape - why?
PD: circle is array of doubles, but shape will have word at start to specifiy the specific type of shape
TR: Is circle being deprecatd?
PD: No, both will coexist. e.g. in SIA, POS takes a shape xtype

GM: Replacing ADQL region
PD: Could replace it currently with shape in obscore - would be easy transition. Another option is to allow services to use either polygon or MOC defined by xtype, but need to think through implications
GM: No STC then, which is good
PD: Yes would be a DALI 2.0 xtype, not STC

Discussion

MD: Case insensitivity mentioned in SIA2 - would make the motion to not do anything case insensitively
PD: Agree, Case insensitivity was a big mistake in earlier SIA so agree

PlS: Would like to have lower function available everywhere. Also pagination is really useful for web presentation
GM: lower is optional in 2.1 but intend to make mandatory (3 mentioned) as they have goot support in DB.
GM: For OFFSET have to be a bit more careful

PD: Would be nice to have a way to deprecate something in a minor version and remove in a later minor version. Likewise introduce something new as optional and then later make it mandatory. Moving things in both directions
MD: Yes agreed. If python can do it in minor versions according to concrete schedule, then we should be able to do some mild evolution. Have already done this for unused functions. Very hard to do major version changes.
DM: What is the problem with a major version change?
TR: Major software (e.g. react) goes with major versions and has breaking changes but used by many more people.
MD: Hard to manage major change in the VO - help out with universal image discovery to see the problem in SIA 1 vs SIA2. Major version change is a huge deal for discovery - splits the universe
DM: In that case why have major and minor versions?

MM: Is it on us to define how to migrate between major versions. But do support deprecations/removals in minor versions. The technology change discussion might force major version changes in the near future
DM: Take the point that it is difficult now. Implies we need to solve that problem. Once we label something as deprecated, what do we gain for removing it? It wouldn't be implemented by new services etc anyway
PD: In code deprecated means you have to maintain it. That may the cost to keeping it. e.g. examples need updating

JT: Are there example implementations of the UDFs for postgres etc?
MD: Yes, implemeted in DaCHS and ESAC. DaCHS is open source. Code is either reliant on pgsphere or preprocessing in python