IVOA October 2022 Interoperability Meeting - DAL Session 1

Time: Tuesday October 18 20:30 UTC

Participants: (52) James Dempsey (JD), Grégory Mantelet (GM), Marco Molinaro, François Bonnarel, Markus Demleitner (MD), Tom Donaldson, Jon Juaristi Campillo (JJC), Mark CresitelloDittmarr, Paul Harrison (PH), Gregory Dubois-Felsmann (GDF), Mark Taylor (MT), Gilles Landais (GL), Daniel Durand (DD), Tamara Civera (TC) et al.

Schedule

Markus Demleitner - ADQL astrometry library: Status update
François Bonnarel - Discovery, description and access of Radio data in the VO. Status report
Tom Donaldson - ADQL and Firewall SQL-Injection Detection
Jon Juaristi Campillo - A proposal for vector math in ADQL

Notes

ADQL astrometry library: Status update

- Folllowing ESAC have two functions with same inputs but different outputs
    - ivo_epoch_prop - return double[6] - proper motions and radial velocities
        - probably the best solution though we do not want to store this 6 values array in a VOTable
    - ivo_epoch_prop_pos - return Point => problem because coords are returned as a string with 2 double in radians
- proposal to have only one `delta_t` argument instead of 2 parameters (`ref_epoch` and `out_epoch`)
- Already possible to test these function in the GAVO DC TAP service
    - differences when comparing result with astropy (in the uas range)
        - neither GAVO nor ESAC apply relativistic correctiosn and secular aberration
- Functions added in pgsphere (code for Lindegren's rigorous treatment ; ESA SP-1200, Hipparcos) ; pull request #8 in pgsphere github (postgrespro)

- Need for a 2nd implementation of `gavo_transform` ; transform coordinates from 1 coord sys to another

MT: agree with Markus on improving the signature of ivo_epoch_prop functions
PH: I prefer from epoch and to epoch - the first epoch tells you when the pm parameters are valid for
GL: VizieR already implements the same function (gavo_transform). VOLLT needs to be improved to accept this function in SELECT

Discovery, description and access of Radio data in the VO. Status report

- Status report on behalf of the Radio Astronomy IG
- PulsarRadioDataAccess in GitHub but in the group ivoa and not ivoa-std - https://github.com/ivoa/PulsarRadioDataAccess
- Obscore extension for Radio data
  - First pass visibility data
  - Expanding to cover single dish data
  - Should this be a note or a recommendation?
  - Comments are welcome on github before it becomes Proposed REC
- Pulsar and FRB (Fast Radio Burst) radio data discovery and access
  - time dependant data
  - File formats including PSRFITS and filterbank
  - Discovery could start from source catalog, Datalink, and/or obscore (e.g. through ObsTAP or DataSet Access Protocol - DAP)
  - Access - could SODA be extended to cover timeseries products
  - Implementation note

GDF: How will the frequency min and max coexist with the mandatory em_min and em_max.
FB: Useful for parameterised services where don't have access to user defined functions (only available in TAP)
GDF: Concerned that these values might represent different situations and not be obvious and potentially the required em values get ignored
FB: These are extra tables so don't replace information, they add information
MD: Yeah, these transformations do belong on the client side.
DD: Meters was selected to be almost waveband neutral, up to the client.
MD: It´s just that in TAP the user interface is on the table level, which is why I'm arguing for the ivo_specconv function.

Brent Miszalski: It may be of interest that we have added MWA (GLEAM-X DR1) data to Data Central's SIA service. We add to each cutout extra header keywords read in from some FITS binary tables that contain position-dependent metadata i.e. info on the beam parameters and completeness levels (from simulations). It may be of interest to some people or worth mentioning in the radio data in VO doc that sometimes extra information is required to be added beyond the usual service descriptions. An example query to look at some cutouts with these headers is https://datacentral.org.au/vo/sia2/query?POS=CIRCLE%2076.4268682%20-28.6702213%200.8&COLLECTION=gleamx_dr1 . Please contact me if you want to know more.

ADQL and Firewall SQL-Injection Detection

- Based on experience that already happened and can still happen
- Whatever client is used (e.g. TOPCAT, PyVO); just adding an ORDER BY clause and TAP query failure due to

connection issue

- WAF/Firewall blocked such queries because they looked like SQL injection
  - solution: add exceptions in the firewall
- But same failures in external services (e.g. GAVO RegTAP service)
  - more network partitions and firewalls all searching for SQL injections ; traffic scanned in all directions
- SQL injection: a way to break out an SQL query to do something else
  - successful attacks quite damaging ; security risks (#3 on the OWASP's top 10)
- Solution proposed by the OWASP Cheat Sheet: use "prepared statements" instead of a building a string
  - quite hard to do with ADQL dynamic queries (especially with ORDER BY value)
  - don't give write access to the DB
  - sanitize user input
  - use firewalls to block SQL injections
- Adding firewall exceptions for specific endpoints possible but is it for > 120 registered TAP endpoints?
- Another solution: changes to TAP standard
  - https for TAP services? may not work forever
  - different encoding of ADQL
  - a different syntax (different from SQL)

MT: https://xkcd.com/327/
MD: agree to adapt TAP and ADQL to cope with such security issue, e.g. allowing ROT13 or the like on the adql string

Tessa Dower: (As an aside, I wanted to thank Gregory again for his work developing and maintaining the ADQL parsing / translation library we rely on at MAST; the safety mechanisms for not exposing unintended tables/columns/etc and general query validation have made it much easier to reassure ourselves as developers, management, and IT security staff as we ask for these exceptions.)

A proposal for vector math in ADQL

- More and more table containing vectors (e.g. Gaia DR3 low resolution spectra)

- Propositions to access to individual elements
    - single element: arr[index]
    - multiple elements with [lower-bound:upper-bound]
- Basic math possible: sum, subtraction, ...
    - overload of math operators (syntactic sugar)
- But also scalar product of 2 vectors : arr_dot(vec1, vec2)
- And array aggregation: avg, max, min, ...
- Usual aggregation operators ok but issues when arrays don't have the same length, ...
- Array not supported by all DBMS and not the same way, but Postgres does
- Missing operations should be implemented with User Defined Functions
    - no `ivo_` prefix yet (clarification by JJC: the idea would be for them to be part of ADQL -- if/when possible)
- As for now, math operations only "fully" supported by GAVO
- Still to be implemented: slicing arrays

JJC: https://blog.g-vo.org/a-proposed-vector-extension-for-adql.html
GDF: Interested because hosting spectra. Would like to take a look on use cases. Is there any plan to have constraints on arrays (in WHERE)?
JJC: yes
TC: saving fluxes in array. Could be a good idea to include that in the ADQL standard.
JJC: would be good to have several implementations especially using other DBMS. Idea of this talk is to propose solutions and see whether there is place for an extension and/or standardisation and I would happy to author them.
TC: http://archive.cefca.es/catalogues/jplus-dr2/help_adql.html -> Functions section