IVOA October 2022 Interoperability Meeting - DAL Session 2

Time: Thursday October 20 06:30 UTC

Participants: (~40) James Dempsey (JD), James Tocknell (JT), François Bonnarel (FB), Marco Molinaro (MM). Thomas Boch (TB), Markus Demleitner (MD), Mark Taylor (MT), Gregory Dubois-Felsmann (GDF), Paul Harrison (PH), Brent Miszalski (BM) et al

Schedule

James Tocknell - User-focused evolution of VO service design
François Bonnarel - DAP, SODA, Datalink status
James Dempsey - Implementing an SIA1 service for CASDA

Notes

James Tocknell - User-focused evolution of VO service design

- Many users do not have a concept of the VO - they are familiar with individual archives
- Presenting more of a 'student' perspective
- Wide variety of software in use - including a lot of older software like IDL and some R
- Some hesitancy to use applications - prefer to script
- data structures may be aligned to specific languages (easier to implement in one language that another)
- OpenAPI https://swagger.io/specification/
- GraphQL https://graphql.org/
- Data Access - more likely to be scripted against by users
- IPC - users are unlikely to upgrade working software to keep up with latest
- Inter service communication - more server to server - less user facing
- Someone needs to write "Accessing data in the VO for Dummies"
- Recent example issues:
    - Astroquery limiting sync to 2000 rows but not advising users
        - Error loudly!
    - Author of DAS having difficulties to interact with different services despite them all implementing the standard, so needed different queries

- DALI - info in VOTable but not in HTTP headers - could potentially use these to allow them to request desired formats
- Recommendations
    - Create non-normative resource aimed at users explaining the service ecosystem
    - Create an implementers and operators guide covering setup;/operation of specific services, useful advice - drive greater similarity - at a level above recommendation
    - Take ideas around service dev process from the Carpentries; lesson devleopment process - "user profiles"
    - Use DA/IPC/ISC to drive design decisions
    - Migration more likely to be held up by users using old tools rather than service providers

TB: old versions of Python are very hard to test in CI
MD: Agree, but are a shoe string business so limits on what can be done.
MD: Have been wanting to do the two docs fofr a whiole - could contribute an hour per month
MT: To help with service interaction - topcat can log web calls using a curl like syntax

François Bonnarel - DAP, SODA, Datalink status

SIA2 -> Data Access Protocol (DAP)

- DAP could do the work of SIA2, SSA2, time series access 1.0, eventlist etc

MD: SDAP, for consistency ("the S-Protocols")
JD: But is it simple?
GDF: “S” = driven by URL query parameters, instead of a query language like ADQL
MT: I agree dropping the "S". TAP was originally going to be STAP - but it hasn't turned out very simple.

DAP Critical points:

- wildcarding
- one shot discovery of cutouts etc
    - SODA-like url in access_ref
    - Service descriptor in query response to build SODA URL

SODA Next

- Errata - MOC parameter, rebin/project, pixel cutout
- Allow selection of outout data type using dataproduct_type
- Cross join of paramters
- Extract metadata

PH: it would be nice to have a “data access query language” to do the SODA parts more flexibly
GDF: SODA-next - would also like a spec for applications like “assemble a temporal cube from cutouts from many single-epoch images”

Datalink 1.1

- Working draft out
- Needs implementations!
- (overview of changes)
- Upcoming imolementations:
    - DACHS - non datasets datalinks
    - CADC -

GDF: DataLink: (now speaking for both Rubin and for Firefly) Very enthusiastic about enhancing the ability to create useful UIs from service descriptors

FB: Param options - min.max discussion - not so mcuh an issue for datalink but DALI/VOtable

GDF: Like to emphasise the rubin/firefly enthusiasm for any datalink content that helps UI building and data entry validation
GDF: Have implemented datalink on top of CADC tap service. System is metadata driven and datalink is central. Using datalink v1.0

James Dempsey - Implementing an SIA1 service for CASDA

CASDA cutout service; from curated datasets, only need is position and radius
Gives back cutouts for overlapping datasets, preview generated on demand
It's an SIAv1 query w/ base URL with custom survey params
The retrieval produces cutouts, sync endpoint is not suitable for large cubes, survey images are kept online to speed up the cutout request.

Why SIAv1?
Needed a survey- rather than image- oriented service
Guide users to the best data
SIA1 support already existing in client, so they are not keen to move on SIA2

GDF: Rubin has the same issue with needing "survey-oriented" cutouts vs. "image-oriented" cutouts. For the coadded data, while the coadds are created in sky tiles, we don't see a reason why users would need to query those first, as the concept of a cutout from the entire coadded dataset is well-defined. So we have been looking at using SODA with a "dataset ID" that corresponds to the entire survey rather than a single coadd tile.

The interim step browse and cut, instead of using the cut directly seems not useful.

Comparison of v1 vs. v2 interaction
v1 is simpler, v2 implies SODA, links and more steps for the same result

Speed hump1: rounding doubles, not implemented in PostgreSQL
Speed hump2: array values through concatenation, what way in ADQL?
Speed hump3: support for SQL VALUES syntax

Discussion on re-building the query URL based on parameters already used in the search phase.
This, while using the links mechanism to actually access (multiform) the dataset (that can be all-sky, large tile, ...)

Discussion on how much to hide to the user and keep on the server side. Solutions to pass on parameters from clients in the datalink URL so that it can include a one-shot cutout row in the datalink results.

BM: We take this approach with our SIA2 service at Data Central
BM: Params passed on to datalink service for the image download urls