

Discussion: IVOA/Astropy/Astroquery/PyVO Collaboration

With the success of the Astropy hack sessions at the 2019 Paris Interoperability Meeting (see summary slides <https://wiki.ivoa.net/internal/IVOA/InterOpMay2019/AstropySprintWrapup.pdf>) and significant VO-related development in PyVO, Astroquery and Astropy since, this is an opportunity to discuss some details of the collaboration to help ensure its continued success.

A draft agenda is suggested below. Please add notes there to help inform and guide the discussions as well as suggesting other potential topics to cover in this or future meetings.

The time schedule is ambitious, so topics may need to be abbreviated (or skipped), but let's try to at least establish the next step for the discussion (splinter meeting, github discussion, etc.).

Agenda

Introduction (Bruno Merin) (10')

A brief high-level review of the goals and recent history of IVOA/Astropy to help set the context for the discussions.

Motivation: Clear synergies between Astropy and IVOA

Goal: Improve collaboration between the groups

History:

Many major archives have modules in Astroquery (Note to Bruno: there is also a NED package in Astroquery)-> Thanks, corrected ! ;)

Great uptake of astropy/astroquery

E.g., usage of astroquery is a large fraction of archive traffic.

Decided in Paris that pyvo is best place for VO code, including TAP client implementations

Questions:

What have we learned in these 2 years? What works and what can we do better?

Where to handle VOTable metadata? E.g., Where to support creation of SkyCoord objects from VOTable sources?

How to unify TAP code?

Tim: General feeling from last 2 years is how to get ESA to contribute code

(speaking of TAP).

How to ensure/increase contributions in general?

PyVO in the last years: New TAP development, SIA2, DataLink, PRs from various groups. When we started it was kind of a dead project, but it is more active now, and we're keeping up with builds, new Astropy versions. CADC and ALMA is using pyvo TAP.

Acknowledging the value of being part of the Astropy family and infrastructure!

Can move more things to pyvo (SAMP, vo_consearch, maybe more)

Where to Handle VOTable Metadata (All - 15')

Astropy has a very solid VOTable parser and writer which is able to recognize and validate some, but not all, of the metadata elements. What has been particularly elusive is doing something meaningful based on the metadata such as creating SkyCoord objects for rows in a source catalog VOTable. With efforts underway in the Data Model working group to make the metadata more rigorous and unambiguous, automatic mapping to Astropy objects may become both more possible and more desired (see the presentations and discussions in the Data Model Workshop sessions this week

<https://wiki.ivoa.net/twiki/bin/view/IVOA/DMWorkshop>). A natural question then is where do such implementations belong (PyVO, Astropy, other)?

Related discussion: <https://github.com/astropy/astroquery/issues/2036>

Related to this, I have started a vo module in Astroquery. The idea is to implement the generic Astroquery API using PyVO components: Reg for discovering the service, TAP for querying it, VOTable, Datalink and SODA to retrieve the data, etc. The module could query multiple services and combine the results. The question of the best home for the module also apply here: Astroquery or PyVO?

Related discussion: <https://github.com/astropy/astroquery/pull/1679>

Other services that are missing:

1. VOSpace client (CADC has a client but requires some effort to be ported over in PyVO)
2. ?

Markus: On question of what can go into pyvo, a pragmatic approach is appropriate. Allow new (experimental, PoC) code in pyvo much like Topcat does. Some metadata is so basic (e.g., coords) it should be in Astropy. Higher level things like interpreting a time series should be in PyVO. But really no general rule

for the middle ground.

Christine: Can have experimental code, perhaps marked as such.

Tim: Different categories on non-standard code: Code that's never expected to be part of a standard, and code that is working towards a (at least potential) standard. Maybe just doc annotations!

MarkCD: Working on this implementation:

https://github.com/ivoa/dm-usecases/blob/main/usecases/standard_properties/mcd-implementation/vizier_pm_anime.gif

but don't know where to go with that code. With a clear statement that pyvo is the place for this, Mark would be interested in being actively involved. More generally, if the IVOA had a more clear path defined for how to develop/integrate prototype code, more developers would be involved, and that experience would make the prototype packages easier to migrate. e.g. The rama package I used (developed by Omar Laurino), is a prototype 'proof-of-concept' type code, and as the annotation question becomes resolved, migrating this into a higher profile package is a natural evolution.

Gerard: Would pyvo be a place for data model implementations, including generator from vodml to those classes.

Laurent: Will present on possible path to interpreting metadata in pyvo.
Thursday May 27 15:00 UTC

Unifying PyVO's TAP with Astroquery's TAPPlus (All - 15')

One of the goals coming out of the Paris meeting was to refactor Astroquery modules to use TAP features implemented in PyVO instead of those implemented in the Astroquery TAPPlus module.

PyVO TAP evolved substantially since then, but is not identical to TAPPlus either in feature set or usability details. How do we move forward towards identifying and implementing the changes needed in PyVO to be able to deprecate TAPPlus and refactor the Astroquery modules that use it?

Related discussion: <https://github.com/astropy/astroquery/pull/2034>

Note that PyVO is already used in Astroquery with 2 modules: alma and cadc (NOTE: nasa_exoplanet_archive also uses it pending PR review). 2 years ago, it was indentified that one main feature that TAPPlus missing in PyVO was **table uploads**. The feature has not been added to PyVO because it's not part of the standard. The discussion here could very well be how to add prototype features to PyVO before changes to the standard are proposed. Or should they be added at all before they make it to the standard?

Christine: No standard table upload. TAPPlus way is good. CADC way is good. No easy way to go from TAPPlus to pyvo when upload is involved. Perhaps refactor table upload into a separate class/module.

Oh, table upload was always there in TAP (though not mandatory). It's persistent uploads that's tricky, and that mainly because that only works with authentication.

Markus: table upload has been in TAP from the start... it's persistent uploads that are tricky.

Mark Taylor : Like Markus, says, tap upload is in the standard.

Jesus Salgado : yes, volatile TAP upload is normal TAP. credentials for persistent upload is the problem

Javier: Can TAPPlus functionality in pyvo?

Tim: We clearly don't want to duplicate features, but where features are distinct, then it makes sense. How can we schedule that work?

Bruno: Somewhat under the impression that non-standard code didn't belong in pyvo. That seems resolved. Other issue was resource limitations especially with Gaia load.

Christine: Really comes down to when will the work be done?

Janet: What's the right communication mechanism?

Christine: E-mail is not the answer. Github issue tracker should be the primary mechanism. The astropy #pyvo slack channel is also useful.

Governance and Process Improvements (All - 10')

- PyVO is currently an Astropy affiliated package, though there were initial plans to work towards a status of coordinated package. Let's clarify the differences and practical impacts and set expectations/goals for future.
- IVOA participation in Astropy projects seems generally good. What are some ways to improve?

Next Steps (All - 5')

Should we schedule splinter meetings or hack sessions to continue work on any topic? What homework should be completed prior to such a meeting?

PyVO/Astroquery Training Links

CDS tutorials

<https://github.com/cds-astro/tutorials>

NAVO Tutorial notebooks:

- <https://github.com/NASA-NAVO/navo-workshop>
- Rendered versions: <https://nasa-navo.github.io/workshop-notebooks/>

Comprehensive PyVO course by GAVO

<http://docs.g-vo.org/pyvo/html/>

Simple PyVO examples (work in progress) by Hendrik

https://github.com/hendhd/pyvo_examples

ADASS 2020 tutorial with jupyter notebooks

https://github.com/hendhd/pyvo_examples

astroquery narrative documentation, with lots of examples for many modules:

<https://astroquery.readthedocs.io/en/latest/>