Astropy Sprint Wrap-Up
IVOA Interop, Paris, May 2019

Tom Donaldson, Erik Tollerud

Sprint Structure

- Small group face to face

- Focused effort for two days

- Communication
    - Circulated topic ideas ahead of the meeting
        - Discussion items
        - Hacks
    - Google Doc for notes and informal proceedings: http://bit.ly/ivoa2019-sprint-doc
    - Slack channel for free form discussion: #ivoa2019-sprint (http://joinslack.astropy.org)

Getting Started

- Introductions

- Add any more items

- Ground rules
  - Astropy Code of Conduct
  - Take notes! (Starting with picking a note taker) - can be in a separate doc linked here
  - Moderate discussions
  - Hacks can be just that: hacky. But still PRed and reviewed if into a regular repo

- Vote on topics

- Split into groups

- Discuss/Hack!

- (Check in once in a while and adjust as needed)

Discussions - TAP

- Multiple organizations contributing TAP-based clients to Astroquery

- What practices might improve collaboration and reduce bottlenecks?
    - Small pull requests can be reviewed more quickly
    - PRs or github issues open communication, so do them early in the process

- Where does TAP client code belong?
    - Astroquery provider-specific modules
    - Astroquery utilities to be used by multiple modules
    - PyVO
    - Somewhere else

- Regardless of final location, Astroquery TAP should be resolved now

- Originally developed by USVAO
  - Astropy affiliated package
  - Low contribution level since; not critical mass for reviews, etc.

- Some renewed interest/use
  - Lingering doubts lead to some functionality being replicated in Astroquery

Consensus reached!

- A robust PyVO would be a good place for VO code
  - Client and server utilities, reference implementations, tools, etc.

- Making PyVO an Astropy coordinated project could have mutual benefits

- Would require:
  - Dedicated maintainers
  - Ongoing contributions (not just solving TAP client)

Hacks – PyVO/TAP

- Evaluate current state of the code
  - Workable, with some very good parts and some issues of various importance
  - See details in doc

- Outline and start next steps:
  - Move PyVO to astropy org (also readthedocs, pypi)
  - Build TAP extensions for authN and persistent table uploads
  - Refactor existing Astroquery modules to use PyVO tools
    - Includes CADC and ESA modules
  - Announce new PyVO
    - IVOA list(s)
    - Astropy dev list
    - Last PyPi release note

Make the metadata accessible when a VOTable is read into an Astropy Table

- What metadata to preserve?
  - Strong desire to avoid complex metadata that doesn't have clear semantics
  - Complicated by multi-table VOTables and unscoped nature of metadata
  - Eventually agreed on preserving all metadata, if it could be done without duplicating the document structure

- How should the end user access the metadata?
  - Structure is complex and semantics unclear
  - For clearer constructs like COOSYS and TIMESYS, assist in creating Astropy objects from Table data

- Hack not completed, but has a clear path forward

- Needed to support VOTable 1.4

Hacks – VOTable character strings

Late started hack: Create strings instead of byte strings for VOTable char array values

- Investigation started
    - Issue clarified
    - Potential solution being tried

Future Plans

- Major progress made!
  - Resolved questions that were blocking progress on several fronts
  - Solid plan for future VO development in Astropy
  - New practices and details learned
  - New connections made

- Let's do this again
  - Likely will be more to do
  - Try to reduce impact on other sessions
  - Try to add interactions beyond a sprint

- Can meet informally at other meetings when there is critical mass