Object Oriented Data Model strategy in the context of IVOA Table Access Protocol services

François Bonnarel¹, Mireille Louys¹,², Laurent Michel³

¹ CDS-France, ² ICUBE-France, ³ ObAS-France

IVOA Table Access Protocol and IVOA datamodels

- TAP (Table access protocol) is an IVOA access standard for astronomical relational databases.
- TAP services registered in the IVOA registry, is widely used and currently expose more than 22 thousands tables.
- TAP makes use of ADQL a universal SQL-like language.
- The database content is described in a standardized schema: the TAP_SCHEMA.
- TAP services built up a query response in one single table.
- Default format for this table is the standard VOTable.

IVOA datamodels:
- are Object oriented
- allow logical description of relationships to data
- can model measurements, coordinates, photometry, complex grouping and provenance
- follow rules defined in the VO-DML standard and are represented in vo-dml-xml format

How do we combine them in practice? ORM solves this only theoretically.
Four possible solutions experimented or considered

Simple flat views (current IVOA standard)

- Simple flat views may be built on top of the data model and the data. In that case the TAP schema covers the needs through standard additional metadata such as ucd, utypes and xtypes.
- Utypes are pointers to the model leaves. Ucd an xtypes express the semantics and format of columns.
- An example is the ObsCore data model.

On the fly TAP response annotation (prototype)

- According to the nature of the datamodel graph (hierarchical or not, with loops or not, etc.) it may be impossible to flatten the datamodel view.
- In that case (Mango, Provenance, etc.) we complete the response with a specific data model mapping syntax currently under development (ModelInstanceInVOTable).
- In our prototype annotation is generated on the fly by the server using some mapping metadata (currently in json) stored in the TAP_SCHEMA.
- The actual mapping depends from the query.
- Mapping client interpreters will retrieve instances of the model and manipulate them as objects.
- See poster X3-010 for details

Ideas for the future: more complex solutions

- A: Renormalized TAP response
  TAP query response are denormalized.
  - It may become strongly redundant.
  - We are considering to use an extension of ADQL which will renormalize the output by adding joins in the multi table response document and qualifying them with help of ModelInstanceInVOTable syntax.
  - This approach will work both for databases fully consistent with a data model (for example Provenance TAP services) as well as for legacy complex catalogs (as in VizieR database).

- B: Full data model instance retrieval
  Model structure may become too complex for on the fly annotation on some provenance use cases.
  - We could create UDF (user defined function) able to retrieve instances of the model in json, yaml, or ad hoc xml, for direct processing by client software.

Acknowledgements: This work was partially funded by the ESCAPE project (https://projectescape.eu/). Additional funding was provided the Observatoire Astronomique de Strasbourg, CNRS and University of Strasbourg.