Fig. 1

Fig. 2

1. TAP in DaCHS

(vgl. Fig. 1)

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

(vgl. Fig. 2)

- What's DaCHS?
- Implementing ADQL
- Implementing TAP
- VOSI and such

2. What's DaCHS?

DaCHS is a multiprotocol VO suite. It supports:

- Most major protocols (SCS, SIAP, SSAP, TAP, OAI-PMH)
- Ingestion from many formats
- Integrated metadata management including resource record generation
- A web interface

It's currently tied to postgres with pgSphere, but we'd like to support sqlite, too.

More (including links to the source) at http://docs.g-vo.org

3. Implementing ADQL

Processing ADQL involves:

- Parsing (to a tree)
- Annotation (of nodes with types, ucds, units, etc)
- Morphing to the RDBMS' SQL
- (Generating VOTables)

4. Parsing ADQL

Since we have EBNF, that's almost straightforward. In DaCHS, we're using pyparsing. However, some rules are written in ways that are unfavourable for recursive descent parsers with limited backtracking, so I had to transform the grammar around some nonterminals like

- table_reference
- column_reference
- joined_table
- unsigned_literal

Details are at the top of adql/grammar.py

5. Annotating ADQL

To figure out names, units, UCDs, and types, you need to annotate column references and expressions in the parse tree.

Name resolution is particularly tricky. My advice: Study the respective passages of the SQL spec thoroughly. I should have done this...

Largely unsolved: Computing UCDs for expressions, inferring units in the presence of literals.

6. Morphing to Postgres' SQL

Apart from geometry, these are relatively harmless modifications of the parse tree (TOP to LIMIT and such).

INTERSECTS/CONTAINS are a bit more tricky since the comparison node must vanish.

Basically, it's another traversal through the annotated parse tree.

7. Generating VOTables

The annotation gives almost all we need to come up with VOTables. Not really solved well so far:

- Embedded metadata (e.g., sources, rights, owners for the tables used)
- Nullvalues

Nullvalues are trouble for integral types since for those, some null value needs to be chosen. Right now, if a null value for a column that has no explicit null value comes up, VOTable production fails. As long as we don't fix the VOTable spec (elsewhere), the solution is to have NOT NULL DDL on all integral DB columns that do not specify a null value in DaCHS' resource descriptors.

8. TAP

There are issues with the upload parameter as discussed on TAPImplementationNotes[1].

REGIONs on input need to be converted to polygons with pgSphere.

UWS job management is done in Postgres tables and using SELECT FOR UPDATE.

9. VOSI and such

There's a bit of confusion as to what should be returned from the VOSI tables endpoint; plus, at some point I'll want a two-stage tables endpoint (because the document is coming into the megabytes).

[1] http://www.ivoa.net/cgi-bin/twiki/bin/view/IVOA/TAPImplementationNotes