



Fig. 1

1. DDC: State of play

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

- Brief reminder: What's this about?
- Service takeover
- Client takeover
- What's missing

Context: Demleitner and Taylor 2016-2019: Discovering Data Collections Within Services, Version 1.1, <http://ivoa.net/documents/Notes/discovercollections>

(cf. Fig. 1)

2. DDC's Job

Discovering Data Collections in Services.

Example: I'm looking for the MACHO archive.

Desired result: Go look in CADC's obscure service (and possibly a few more places).

Technique: "auxiliary capability" in a suitable record on MACHO pointing to CADC's TAP service.

3. Server-Side Takeup

isServedBy relationship to main service: Found in 29 registries.

Try it yourself:

```
select distinct harvested_from
from rr.relationship natural
join rr.resource
where relationship_type = 'isservedby';
```

(harvested_from is a local addition on reg.g-vo.org). There are a few relationships in there that are non-DDC, but only very few.

tap#aux capabilities: Found in 27 registries, among them Vizier, Heasarc, Esavo.

Services need to have tabledata metadata in their Registry records:

46898 tables in rr.res_table

48240 tables in glots.tables.

4. Missing Tables

What would we miss if TOPCAT's TAP discovery switched to RegTAP now?

That's the reason for the GloTS comparison: anything in GloTS but not in rr will get lots. That's ~ 1350 tables at this point. Here's a query that lets you figure out roughly what it is that disappears:

```
select ivoid, table_name
from glots.tables natural
left outer join rr.res_table
where schema_index is null;
```

This includes Simbad, CADC, IRSA, ESAVO, Edinburgh, INAF, CASDA... Ouch. Can't switch just yet. Of course, the query above is a bit pessimistic, because it assumes that people use fully qualified table names in both TAP.SCHEMA and their VODataService tableset. That's not been explicitly required so far on both.

5. What can we do?

Just insert your VOSI table endpoint's output into your TAP registry record (better: assign tables to the resource with the aux capabilities).

It would also help if the web-based registries at STScI and ESAVO could start checking for VOSI tables endpoints for services registered with them.

6. Client Takeup

Both TOPCAT and Aladin base their default discovery on GloTS still.

TOPCAT's RegTAP-based discovery mode doesn't look at rr.res_table yet and hence works very differently from GloTS mode.

Aladin queries aux capabilities, but creates a separate plane for each even if they all end up at the same service.

I'm happy to work with client authors to see how to improve this.

7. Oh: Why Bother?

Currently: TAP discovery uses GloTS, which is non-standard and, realistically, depends on me.

So, if you don't want DDC, go standardise GloTS.

In the future: As we go into the 10^4 range of published data collections in SSAP, SIAP, Obscore, we can't have one service per data collection *and* all-VO queries.

Well: I'd go for keeping all-VO queries any day.

8. Encore: Obscure

Problem for Aladin: There's a data collection (DC) with a TAP aux capability.

How can it figure out it's supposed to look in obscure?

- Have `ivoa.obscure` in the tableset of the DC? That's a *lot* of columns for that one single fact, and these columns are probably already declared on the main service.
- Perhaps an empty table head of `ivoa.obscure` would do?
- Have a separate obscure resource record and make a relationship there? We'd have to define how this obscure resource would look like then. Hm. Also, Aladin would have to dereference all these relationships and see if one of them is such an obscure resource.
- Have a `VOResource` type for auxiliary capabilities that communicates extra metadata? That could also help to communicate arguments to select only a specific subset of data in collection S*AP services. I, for one, would like to think more about where this idea leads to.
- Anything else?

Ideas welcome!