Fig. 1

1. Takeup of New Registry Tech

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

New Registry Tech as in

- DOIs and such (VOResource 1.1), for systematic citability
- STC coverage (VODataService 1.2), for blind discovery
- Table sizes (VODataService 1.2), for blind discovery
- Table registration (DDC), to get rid of GloTS

(cf. Fig. 1)

2. DOIs for Registry Records

I'd like it a lot if clients could say something like "cite DOI suchandsuch if you use this data."

However:

```
select count(*) from rr.alt_identifier
where alt_identifier like 'doi:%';
```

yields 56. That won't cut it for client takeup.

Want a DOI but don't know where to get one? Try voidoi[1]!

[1] `http://dc.g-vo.org/voidoi/q/ui/custom`

3. STC Coverage

Wouldn't it be cool to be able to say "I'm looking for Hα in M8"? VODataService 1.2 would let you do that, but:

- 15131 resources have spatial coverage, but 99% are pulled from footprints, so it's cheating.
- 80 have temporal coverage (5 authorities)
- 75 have spectral coverage (4 authorities)

Again: Too lame to call for client takeup.
Add STC info! It's not really hard.

In case you're wondering what queries I ran to come up with these numbers – since a single resource can have multiple records in the temporal and spectral table, a little care is necessary when counting them:

```
select count(*) from (
        select distinct ivoid from rr.stc_temporal) q
```

To count the authorities, you need a DaCHS-specific user defined function (or so I think):

```
select count(*) from (
        select distinct gavo_getauthority(ivoid) from rr.stc_temporal) q
```

4. Table Size

VODataService 1.2 also adds an `nrows` attribute to table definitions. Use case: "Find a large catalogue", perhaps as a sort criterion.

There's 51 resources with nrows right now.

Not mapped in a RegTAP service yet. I'll do a prototype if *you* put @nrows into your records.

Actually, I'm not including nrows in many of my tables. That's because I've noticed that the way DaCHS does this is incredibly clumsy and painful, and it needs to change. But that won't happen before DaCHS has been ported to python 3. Holding one's breath is probably not a good idea at this point.

5. Table Registration

Here, we want all tables in GloTS (49209) in rr.res_table (48158), too. To make discovery work, that then requires auxiliary capabilities.

Looks like we're almost there. But it's not that easy.

Figuring out what tables come from where isn't quite straightforward in RR because there's both primary and auxilary services – this might be an indication that a nice view is in place. Here's what I ran for a count of tables per service:

```
with aux as (
  select related_id as ivoid, table_name from
    rr.res_table
    natural join rr.capability
    natural join rr.relationship
  where
    standard_id='ivo://ivoa.net/std/tap#aux'
    and relationship_type='isservedby'),

alltables as (
  select ivoid, table_name from
    rr.res_table
    natural join rr.capability
  where standard_id='ivo://ivoa.net/std/tap'
    union
    select * from aux)

select ivoid, count(*)
from alltables
group by ivoid
```

The corresponding GloTS query is rather non-scary:

```
select ivoid, count(*)
from glots.tables
group by ivoid
```

From the large services, more than 50% of IRSA is missing, all of esavo/gaia, esasky, ARI-GAIA, ...

Not quite ready for prime time, but with client support, it shouldn't be hard to fix what's left. Let's go!