



Fig. 1



Fig. 2

1. Registry Developments

(cf. Fig. 1)

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

(cf. Fig. 2)

- SimpleDALRegExt 1.1
- TAPRegExt 1.1
- DOIs
- STC searches in the Registry
- Registering data collections

2. SimpleDALRegExt 1.1

Why?

How do I register my SIAPv2 service?

□ SimpleDALRegExt 1.0 fixes the @standardID of a SIAP capability to the old, SIAP version 1 value. Thus, SIAPv2 cannot use the registry extension, and thus, services cannot communicate test queries.

□ SimpleDALRegExt will essentially just lift that restriction, so you can say:

```
<capability standardID="-ivo://ivoa.net/std/SIA#sync-2.0"
  xsi:type="sia:SimpleImageAccess">
  <interface role="std" xsi:type="vs:ParamHTTP">
    <accessURL use="base">http://dc.zah.uni[...]
  </interface>
  <testQuery>
    <pos>
      <long>230.444</long>
```

[...]

I'm not so hot about any other SIAP metadata.

Are you?

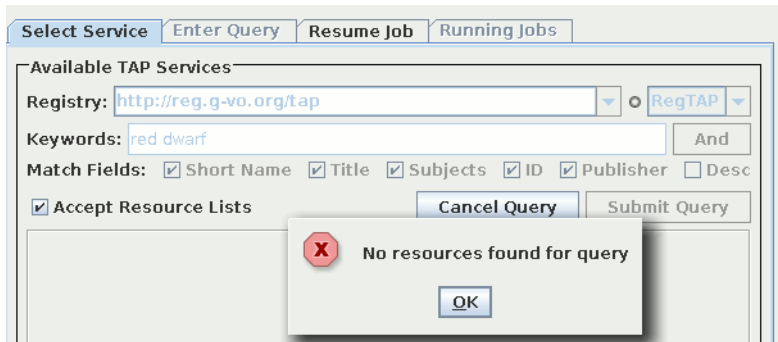


Fig. 3

3. TAPRegExt 1.1

TAP 1.1 will have a problem just like SIAPv2.

TAPRegExt 1.1 solves it in exactly the same way.

And fixes two spec bugs.

Is Demleitner et al, 2012 really perfect otherwise?

4. DOIs

I've written a web service for minining DOIs from IVOIDs (but it's switched off so far, and tombstoning is not implemented).

Challenge: We need to update VOResource to support altIdentifier and possibly make it a bit more DataCite compatible.

I simply didn't get around to do it. Help!

5. STC searches in the registry

The plan still is: Dump MOCs into a database table. More on this by Markus N.

6. Registering Data Collections

There's a Note on this now: Note 2016-01-08¹

Problem: Letting clients find TAP tables by metadata:

(cf. Fig. 3)

- But this isn't only about TAP. Similarly, SIAP or SSAP services could be a common frontend
- for archives of several instruments or observatories.

¹ <http://ivoa.net/documents/Notes/DataCollect>

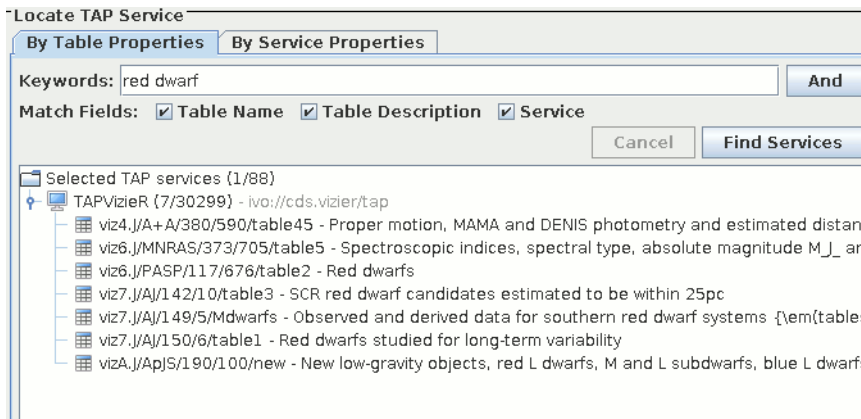


Fig. 4

7. But isn't it fine now?

(cf. Fig. 4)

- ⌋ ...except it's using GloTS (i.e., a join of all TAP_SCHEMAS). And that's bad because it only
- ⌋ works for TAP and has a different, much poorer metadata model than the Registry).

8. Note Proposal

Data collections get a CatalogService record with

```
<capability standardID=
  "ivo://ivoa.net/std/TAP#sync-aux-1.1">
  <interface xsi:type="vs:ParamHTTP" role="std">
    <accessURL use="base">(service access URL)<accessURL>
  </interface>
</capability>
("auxiliary capability") and
<relationship>
  <relationshipType>served-by</relationshipType>
  <relatedResource ivo-id="(main service IVOID)"
    >(terse name of the main service)</relatedResource>
</relationship>
```

9. Two use cases

(1) Service enumeration: "Give me all TAP 1.1 services so I can query everything there is in the VO."

Match against `ivo://ivoa.net/std/ssa#tap-1.1`.

(2) Resource discovery: "Are there TAP 1.1 services that publish data on RR Lyr stars?"

Match against `ivo://ivoa.net/std/ssa#tap-aux-1.1`.

(plus of course additional constraints).

With SQL patterns, broader version subsets can be selected.

10. A Dirty Little Secret

I want this to be applicable immediately. Therefore, the Note defines the aux capability ids

- `ivo://ivoa.net/std/conesearch#aux` (SCS 1.03)
- `ivo://ivoa.net/std/sia#aux` (SIAP 1.0)
- `ivo://ivoa.net/std/ssa#aux` (SSA 1.*)
- `ivo://ivoa.net/std/tap#aux` (TAP 1.0)

– which work as proposed, but defining standard keys in other standards might be considered an unfriendly act.

11. Discussion

Well...?