Fig. 1

Fig. 2

1. How to Register Your Data

(cf. Fig. 1)

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

(cf. Fig. 2)

- Why register?
- Registry architecture
- Web-based registration
- Less Manual alternatives

2. Why Register?

Google (or any other generic search engine) is lousy if you're hoping to find answers to queries like:

- Give me all infrared image services
- Give me catalogs of quasars with redshifts
- Who can I contact if the service at http://foo.bar/svc is down?
- What services are there service Planck data?

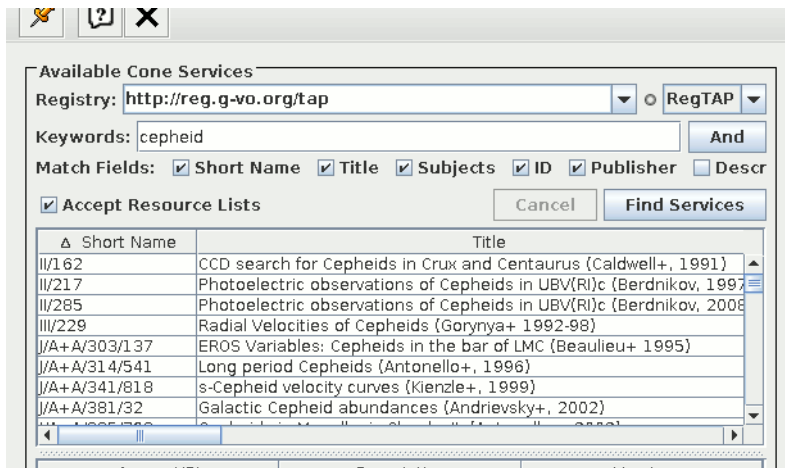To tackle these, we need reliably structured data.

Fig. 3

3. Also, Register to. . .

. . . be in TOPCAT:

(cf. Fig. 3)

4. Registry Architecture I

∩ The VO defines a metadata model for services and a few other things we need metadata for. The basic standard saying how that works is called VOResource, and there are several extensions to that (VODataService with services, SimpleDALRegExt for data services with standard VO
∪ protocols, TAPRegExt for TAP services. . .). A VOResource record for a service contains:

- Title, identifier

- Curation info (publisher, contact, version. . .)

- Content info (description, keyword, bibliographic source, reference URL, relationships to other resources. . .)

- Capabilities (ways to access the data: web page, standard services. . .)

- Coverage (of the data contained in space, time, spectrum)

- Tableset (structure of the underlying table)

□ If you look closely, there's yet a bit more, but that's the rought structure.

Fig. 4

5. Registry Architecture II

In the VO, service operators describe their resources themselves (they produce "Registry records" and put them into "publishing registries"). To be searchable, they in effect need to be concentrated in one place (a "searchable registry").

(cf. Fig. 4)

6. Registry Architecture III

The Registry has a primary key, the identifier:

$$\mathtt{ivo}://\underbrace{\mathtt{dc.gavo}}_{\text{authority}}/\mathtt{whatever/resource/path}$$

Identifiers must be unique:

- for the authority, the Registry guarantees uniqueness (by requiring that you register your authority and not letting you do that if the authority is already taken)

- for the rest, it's your job (but you'll only clobber your own records if you mess up)

Fig. 5

Fig. 6

7. Registering Your Service

If all you have is a service or two, possibly just browser-based, there's no need to run a publishing registry yourself.

EuroVO runs one for you:

```
http://registry.euro-vo.org/eurovo/
```

First step: Grab your authority (Hint: institute acronym plus country or project name is probably good).

(cf. Fig. 5)

8. Registering Your Authority

It then asks for a bit of extra metadata.

Reference URL is where to find out more about your institute.

Unless you have good reason not to, choose

- Publisher=institute
- Title = Publisher + "Naming Authority"
- Managing Org = Publisher

∩
∪ Sorry about the lack of space in description – it should be about a paragraph about what names you expect to be handing out.

(cf. Fig. 6)

After completing your registration form (which essentially creates the authority for you), you'll have to wait a bit because your data is checked manually.

Oh, and don't worry: Unless you make something visible explicitly, on the EuroVO site it remains hidden, i.e., it's not visible to the Greater VO yet. This includes your authority, so when you're here, you can still abort.

When your account is approved, please make sure you edit your record and set contact/email before publication – when something is wrong with one of your records, this is where Registry maintainers want to go, and giving them an easy means to reach them is nice.

9. Registering Your Resource

Demo time for

http://dc.zah.uni-heidelberg.de/antares10/q/cone/form

Here's a bit of guarding rail to help you through with what I've done.

1. In the sidebar, select the "My Resources" icon. Hit the "Add new resource button" on the resulting screen.

2. Select the "Use wizard" radio button in the resulting dialog, and, for a web page with data, "Catalog Service 1.0".

3. For ivoid, make up something that somehow fits concisely what your service is about. Use hierarchy to reduce collision risk.

4. Publisher is again probably you

5. For subject, try to find something in the IVOA Thesaurus[1]; add more keywords later, just have one here.

6. Reference URL could be something like a help page, or the landing page of a publication. If nothing else works, use your web page.

7. Hit Ok, but keep editing in the resulting form – your record still stinks.

8. Make a ShortName (in the first column in TOPCAT and similar interfaces, so make it pretty by less than 16 chars).

9. Set the data authors in creator; the "URL" below lets you put in a logo – if you put in one, make it a smallish file.

10. In Date, you could add arbitrary dates relevant to your content; most usefully, you could use "update" as role.

11. Again, pleaseplease add your e-mail address in Contact

12. In source, have format=bibcode and give the bibcode of a publication with more info (which is also what people should cite).

13. Don't sweat Type and Content Level too much – that's not really used for discovery right now.

14. Relationship is for advanced uses mainly. Don't just make vain relationships to indicate that, perhaps, your institute publishes the service – that's what publisher is for.

15. Rights isn't too useful right now either; set it anyway

16. Most important: "Add Element" for Capability. For a web page, select plain Capability ("vr:Capability").

17. You don't need to add anything to the capability except an Interface. Use the vs:WebBrowser type. No role or version need to be given.

18. Finally, add the URL of your web page (as access URL)

---

[1] http://www.ivoa.net/rdf/Vocabularies/vocabularies-20091007/IVOAT/dict/A.html

```
<service id= cone  core= xtcore  attoweu= rorm,scs.xmt >
    <publish render="scs.xml" sets="ivo_managed"/>
    <publish render="form" sets="local,ivo_managed,gavo"/>
  <meta name="title">PPMXL Cone Search</meta>
```
Fig. 7

19. Hit the disk item so save the record

20. Check if everything looks good, then hit the eye icon to publish the record ("point of no return")

21. It'll take a day or so for your service to show up in the usual publishing registries.

If you create records for actual standard wervices, you can or must add additional metadata in the capability. And you can immediately validate whether you implement the protocol correctly.

10. Alternatives

You can run a publishing registry yourself. GAVO DaCHS, for instance, makes that reasonably easy. After that, from a DaCHS service description, you can say:

(cf. Fig. 7)

Or: produce your own resource record XML and put it somewhere where we or EuroVO can find it.

This could be the preferred way for medium installations. We'd have to write it, though. Talk to me if interested.

11. Even if. . .

. . . you're not running a VO standard service yet. . .   . . . register it!
Thanks.