



Fig. 1



Fig. 2

## 1. Towards Registry Interfaces 2

(vgl. Fig. 1)

Markus Demleitner  
msdemlei@ari.uni-heidelberg.de

(vgl. Fig. 2)

- Why Registry Interfaces 2?
- Brief RegTAP overview
- What's new in RegTAP?
- Requests for discussion

## 2. Why Registry Interfaces 2?

- RII relies on several deprecated technologies (an ad-hoc protocol on top of SOAP, ADQL version 1 in XML serialization).
- RII underspecifies what's being queried against, so even fairly simple queries behave rather differently on the various registries, complex queries appear to be impossible. The net result is that the VO's registry infrastructure is dramatically underused; that's particularly bad since it's probably going to be what will enable the VO killer application.
- The RII infrastructure is in need of lots of work and maintenance; re-using existing components (e.g., TAP servers or Solr) will be less work in the medium run.

## 3. Registry Interfaces 2

There's an internal working draft<sup>1</sup> on volute, consisting of:

- The IVOA Harvesting Interface (OAI-PMH plus some restrictions, only redactional changes so far; well, there's some relaxation on the rules with deleted records, but that's not a big deal).
- "Searching the Registry", a complete rewrite based on TAP ("RegTAP" in parallel to "ObsTAP").
- Registering Registries (basically unchanged, except we don't need to register searchable interfaces any more, TAPRegExt does it for us).

## 4. RegTAP in Brief

Basic RegTAP is 13 tables mapping most of VOResource. These were discussed in some breadth in Urbana, so here's the short version –

- resource (what's 1:1 to an IVORN, plus a few #-concatenated values)
- res\_role (creators, publishers, contacts)
- capability, interface, intf\_param (access metadata incl. query parameters, where the access URL is folded into interface)
- res\_schema, res\_table, table\_column (that's basically VODataService)
- subject, res\_date (res\_date is for curation/date and might be another candidate for absorption into res\_detail, provided we generated funky utypes here)
- relationship
- validation
- res\_detail (that's utype-value pairs for registry extension and arcane features)

## 5. What's new in RegTAP?

Compared to what I showed in Urbana, here's the main changes to the relational registry schema:

- There's now a single res\_role table that combines the previous creator table and the publisher and contact fields from the resource table. This lets you have multiple contacts again.
- accessurl is folded into interface so there's only one access URL per interface now
- resource.creator\_seq keeps creator ("author") names in the source sequence for presentation purposes
- capability\_detail is now res\_detail (which still has a cap\_index column; if that is NULL, it's a detail on the whole resource now. The spec now has a table of utypes for the registry extensions known to date.)
- resourcevalidation and capabilityvalidation became a single table with a convention as for res\_detail to tell resource from capability validation
- New user-defined functions: ivo\_nocasematch, ivo\_hasword, ivo\_hashlist\_has.

<sup>1</sup> <http://volute.googlecode.com/svn/trunk/projects/registry/RegistryInterface/RegistryInterface2-fmt.html>

## 6. The STC Extension

There's now a separate "Data Model" with IVORN `ivo://ivoa.net/std/RegistryInterface#stc`. It consists of tables like:

<code>ivoid</code>	<code>vor:resource.identifier</code>
<code>time_start</code>	<code>stc:AstroCoordArea.TimeInterval.StartTime</code>
<code>time_end</code>	<code>stc:AstroCoordArea.TimeInterval.EndTime</code>

for space, time, spectral, and redshift. This means: service coverage is modelled as a union of intervals (or bounding boxes).

For space, we initially planned to have a column containing an ADQL geometry to allow complex shapes. That's very hard without polymorphous geometries in the backend database since a geometry type that can represent both AllSky and weird polygons is nontrivial, as is splitting up large coverages to turn them into unions of smaller shapes.

Figuring out bounding boxes is, in comparison, reasonable, even on a sphere.

## 7. Calls for Discussion

### Do we need this?

Well, we need *something*, and quick. This is actually rather cheap, since we get TAP and ADQL for (almost) free. If you already have a harvester and OAI-PMH, this is about 500 lines of python, probably a bit more in XSLT. And it can do all the use cases in the Wiki<sup>2</sup>.

### Do we want all role-like things in one table?

I'm neutral here, except that we should have "creator" easily searchable (on ADS, by far the most queries involve authors).

### Should the RR tables live in the ivoa schema?

The ivoa schema was pioneered by Obscore; of course, obscure only defines a single table. So, I think we're fine claiming a new schema.

## 8. More Discussion

### Should we require a table containing VOResource XML?

This certainly would be useful for presentation, and if you're running OAI-PMH you'll probably have such a table anyway. On the other hand, the details (namespace mappings come to mind) are ugly.

### Utypes

Absent VOResource in VO-URP or whatever else, we're free to beautify the utypes we use. It would be nicer to have `vor:standard.endorsedVersion` rather than `vor:resource.endorsedVersion`. But we certainly don't want `vor:standard.curation`, so it's tricky.

## 9. Still more discussion

### Do we want to talk about STC at all?

I think we should. Even if STC-X sucks, if we want to keep all-VO searches feasible we simply *have* to filter resources at the registry level, and coverage is an obvious and fairly powerful filter. The next VOResource should limit the STC-X constructs available, though, or define some coverage elements itself.

### Who's gonna implement this? How many of these are going to be around?

Well, *that* is the question. Try it. It's not terribly hard.

We'll have a second, geographically distributed server for this stuff soon-ish, and I hope to win some of the DaCHS deployers. Still, there should be independent implementations.

<sup>2</sup> <http://wiki.ivoa.net/twiki/bin/view/IVOA/RestfulRegistryInterfaceReq>