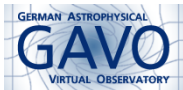


# RegTAP: A Relational Registry for the VO

Markus Demleitner, Florian Rothmaier

Astronomisches Rechen-Institut Heidelberg

January 16, 2013



# Why RegTAP?

- the Registry Interface 1 (“RI1”) relies on **deprecated technologies**
  - ▶ SOAP-based web services based on XQuery and on ADQL 1.0
- RI1 **underspecifies** whats being queried against
  - ▶ even simple queries behave in a rather different way on the various registries
- the RI1 infrastructure is in need of **a lot of work and maintenance**
  - ▶ it seemed to be less work to use existing components like TAP servers and ADQL 2.0 in the medium term

- internal working draft on volute<sup>1</sup>
- the interface for **searching registries**: a complete rewrite based on TAP (“RegTAP”)
  - ▶ RegTAP consists of **13 official tables** plus four tables for the STC extension (wish list)
- **three ADQL user-defined functions** for pattern matching, string comparison
- design guidelines: our goal was not to entirely cover the VOResource schema but to provide a **robust and user-friendly interface**

---

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

# The resource Table

- contains most atomic members of `vr:Resource` (type defined in the `VOResource` schema)

Name	UType
<code>ivoid</code>	<code>vor:resource.identifier</code>
<code>res_type</code>	<code>vor:resource.type</code>
<code>created</code>	<code>vor:resource.created</code>
<code>res_title</code>	<code>vor:resource.title</code>
<code>updated</code>	<code>vor:resource.updated</code>
<code>reference_url</code>	<code>vor:resource.content.referenceURL</code>
<code>...</code>	<code>...</code>

- given UTypes refer to the `VOResource` schema equivalent
- primary key of the table: `ivoid` (all tables have this column)
- further table columns: information about the creator of the resource, the coverage, the waveband, the instrument, ...

# The interface Table

- this table subsumes the attributes of the various VOResource interfaces

<b>Name</b>	<b>UType</b>
<code>ivoid</code>	<code>vor:resource.identifier</code>
<code>intf_type</code>	<code>vor:interface.type</code>
<code>res_title</code>	<code>vor:resource.title</code>
<code>query_type</code>	<code>vor:interface.queryType</code>
<code>result_type</code>	<code>vor:interface.resultType</code>
<code>access_url</code>	<code>vor:interface.accessURL</code>
<code>...</code>	<code>...</code>

- unlike the VOResource schema, RegTAP only allows one URL per interface
  - ▶ queries simplified by this design decision

# The res\_details Table

- this table gives utype-value pairs for registry extensions

Name	UType
ivoid	vor:resource.identifier
cap_index	
detail_utype	
detail_value	

- in particular this key-value mechanism is intended as an easy fallback for new VOResource extensions
- example: “Find all TAP endpoints offering the relational registry.”

# The res\_details Table

- this table gives utype-value pairs for registry extensions

Name	UType
ivoid	vor:resource.identifier
cap_index	
detail_utype	
detail_value	

- in particular this key-value mechanism is intended as an easy fallback for new VOResource extensions
- example: “Find all TAP endpoints offering the relational registry.”

```
SELECT access_url
FROM rr.interface
  NATURAL JOIN rr.capability
  NATURAL JOIN rr.res_detail
WHERE standard_id='ivo://ivoa.net/std/tap'
  AND detail_utype='vor:capability.datamodel.ivoid'
  AND detail_value='ivo://ivoa.net/std/regtap/vor'
```

- capabilities: descriptions of what a resource does and how to use it
- res\_role: information about creators, publishers and contacts
- intf\_parameters: description of the input parameters that can be provided to the service as a name=value argument
- res\_schema, res\_table, table\_column: basically VODDataService (metadata on columns of tables / descriptions of tables pertaining to the resource)



- `capabilities`: descriptions of what a resource does and how to use it
- `res_role`: information about creators, publishers and contacts
- `intf_parameters`: description of the input parameters that can be provided to the service as a `name=value` argument
- `res_schema`, `res_table`, `table_column`: basically `VODataService` (metadata on columns of tables / descriptions of tables pertaining to the resource)
- `subject`: keywords
- `res_date`: date associated with an event in the life cycle of the resource
- `relationship`: descriptions of relationships between resources
- `validation`: validation levels for resources and capabilities

# On Our Wish List: The STC Extension

- separate “Data Model” with IVORN  
`ivo://ivoa.net/std/RegistryInterface#stc`
- many use cases for this extension
- four STC tables in RegTAP: `rr.stc_spatial`, `rr.stc_temporal`, `rr.stc_spectral`, `rr.stc_redshift`
  - ▶ e.g. table `rr.stc_temporal`

Name	UType
<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>

➔ service coverage is modelled as a union of intervals (or bounding boxes)

- “Find all SIA services that provide infrared images.”

```
SELECT ivo_id, access_url
FROM rr.capability
      NATURAL JOIN rr.resource
      NATURAL JOIN rr.interface
WHERE  standard_id='ivo://ivoa.net/std/sia'
      AND waveband LIKE '%infrared%'
```

- “Find all SIA services that provide infrared images.”

```
SELECT ivoid, access_url
FROM rr.capability
     NATURAL JOIN rr.resource
     NATURAL JOIN rr.interface
WHERE standard_id='ivo://ivoa.net/std/sia'
     AND waveband LIKE '%infrared%'
```

- “Find all TAP services; return their accessURLs.”

```
SELECT ivoid, access_url
FROM rr.capability
     NATURAL JOIN rr.interface
WHERE standard_id='ivo://ivoa.net/std/tap'
```

# Conclusion

- our system is running
  - ▶ in Heidelberg and in the very near future on a mirror in Potsdam
- the ESA will adopt our system pretty soon
- TAP client libraries are already available
- registry web interface planned by GAVO Heidelberg

➔ try it and consider updating your own clients



K. Benson et al.: IVOA Registry Interfaces Version 1.0, IVOA Recommendation (2009-11-04),

<http://www.ivoa.net/Documents/RegistryInterface/20091104/REC-RegistryInterface-1.0.pdf>.



M. Demleitner et al.: IVOA Registry Relational Schema, Version 1.0, IVOA Working Draft (2012-11-12)

[http:](http://volute.googlecode.com/svn/trunk/projects/registry/regtap/RegTAP-fmt.html)

[//volute.googlecode.com/svn/trunk/projects/registry/regtap/RegTAP-fmt.html](http://volute.googlecode.com/svn/trunk/projects/registry/regtap/RegTAP-fmt.html)



M. Demleitner: A relational model for VOResource, talk given at the IVOA interop in Urbana (May 2012).



M. Demleitner: Towards Registry Interfaces 2, talk given at the IVOA interop in Sao Paulo (October 2012).