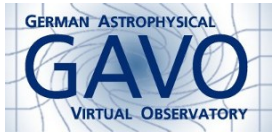

*Fig. 1*

*Fig. 2*

# 1. The VO And Why It Matters

(cf. Fig. 1)

Markus Demleitner
*msdemlei@ari.uni-heidelberg.de*

(cf. Fig. 2)

- What is the VO?
- The VO's answers
- What can you do?
- Publish data or let it perish



*Fig. 3*

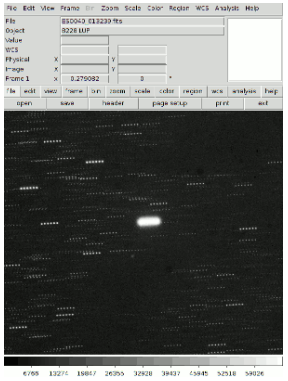
*Fig. 4*

# 2. What's the VO?

The Virtual Observatory is **not**...

(cf. Fig. 3)

...a platform or web page – as that would restrict what you can do essentially to what the platform operator thought you should be able to do. There's nothing wrong with having VO clients run in a web page, though.

(cf. Fig. 4)

...a programme – you *use* programmes to access the VO, but the programmes typically do other things (process tables, work with images, analyse spectra...)

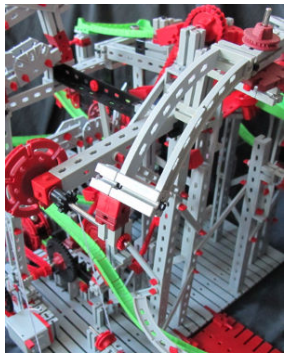

*Fig. 5*

(cf. Fig. 5)

. . . a bizarre contraption – you can do a lot of exciting research with the VO, and these days it becomes harder and harder to do astronomical research without the VO (though you may not notice it).

## 3. What's the VO?

The Virtual Observatory (VO) is (or will be), a

**comprehensive** set of
**data** and **services**
relevant to **astronomy**
accessible from **clients** of **your choice**
**regardless of where you are** and
**preserving** products of digital astronomy.

## 4. "comprehensive"

The VO intends to allow access to basically all astronomical data, present and past.

Right now: About 20000 "resources" like

- VizieR catalogs
- Lots of space missons
- Many observatory collections
- Theory data like synthetic spectra
- Much more

But there's still a bit missing.

VO jargon: A "resource" essentially is "something that has metadata". Some of these the VO uses internally, but most of them really are what the next slide discusses as "data and services".



*Fig. 6*

## 5. Data and Services

While the VO is about data, much of it is concerned with services.

A service is

- a piece of software accessible via a network
- with a well-defined interface
- allowing access to some data collection.

The tricky bit: Service users ("clients") must be able to figure out how to operate the service and find out as much as possible about the data contained. This is so people can automate things operating over multiple service – and that, in turn, is a precondition to more cross-resource research.

## 6. Astronomy

Well, of course. . .

(cf. Fig. 6)

. . . but other fields have similar endeavours, and they're using similar technology (OAI-PMH, SQL. . . ).

## 7. Clients and Choice

"Web pages" aren't really what the VO is about. It is about standard interfaces to data.

This means: A single programme (possibly web-based) can operate all kinds of archives and services. Many such programmes are listed at http://ivoa.net.

It also means: A given service can be operated by any client speaking the VO languages – you get to choose or use libraries like pyVO in your own programmes.

## 8. The big equalizer

It used to be that you had to go to the big observatories to get top-notch data.

Converseley, chances someone would see and use your data if you weren't there weren't terribly good.

The VO already delivers excellent data to anywhere and anyone in the world.

And with All-VO searches and increased adoption of Registry use, everyone gets a more uniform view of the data taken.

## 9. Preservation

We're currently losing historical observations at an unprecedented rate: All the tapes from the 80ies and 90ies are deteriorating.

Linus Torvalds:

> Only wimps use tape backup: real men just upload their important stuff on ftp, and let the rest of the world mirror it.

If data is to survive, it must be in living services not far from spinning disks.

(Yes, there's more to it, but the living part is vital)

## 10. VO Reality

To make this nice "comprehensive set" useful, it must let you

- Find data relevant for your research,
- Get it, and
- Use it

Compare to literature: **Find** a paper on ADS, **get it** using a web browser ("client") from a publisher's web page (or, if you're lucky, from ADS itself), **use it** in your PDF reader. VO jargon: A dataset is understood to be an "individual data item with included metadata", which could be a table, a spectrum, an image, a data cube, or yet something else. Since a set of such things needs a name, too, and dataset is not available, we call that a data collection.

Also, maybe the word "metadata" deserves a brief comment: Metadata is "data on data". For an image, that could be "When was it taken?", "What filter was used?", "Where does it point?", "What does it show?", etc.



## 11. The VO way

In theory, all those data collections could reside in one, professionally managed place.

This would be like ADS; the publishers deliver their data, and the ADS staff unifies and "curates" this.

In reality, such a place doesn't exist. Although for tabular data, VizieR comes pretty close.

**The VO way**: Let there be many data centers, but have them speak common languages ("protocols") and make it so their metadata can be collected and interpreted by machines.

This is a bit like the Web, where there's lots and lots of web servers, but google's robots can harvest what's on them and provide an index (only there's more webservers and far less structure in the Web).

## 12. Finding Services

The union of the metadata of all the data centers in the VO is called the **registry**. There, users can issue queries like:

- Where are image services specialized on radio?
- What data sets are out there containing x-ray fluxes and proper motions?
- What services are out there dealing with time standards?
- What services expose the data associated to a paper?

Clients: WIRR[1], NAVO Directory[2], In-Application interfaces, pyVO. You can also query the registry using the TAP/ADQL clients mentioned below using the TAP access URL http://dc.g-vo.org/tap. If I (as the author of the respective standards) may say so, this is probably the way to go if you're planning advanced stuff with the registry.

1 `http://dc.g-vo.org`
2 `http://vao.stsci.edu/keyword-search/`

## 13. Finding Data Sets

The VO has defined "**typed interfaces**" that let you talk to all services in the same fashion. "Typed" means literally types of data. There is, for example, "Simple Cone Search" (SCS) for tables with sky positions in them, the "Simple Image Access Protocol" (SIAP) dealing with images of the sky, and "Simple Spectral Access Protocol" (SSAP) for accessing spectra.

The common language lets programmes query many servers at one click. So, you can ask questions like:

- Find all images containing NGC3141
- Are there infrared spectra of a source at 271.8281, +23.42?
- What is known about sources within 2 arcminutes of Geminga?

Clients: TOPCAT[3] for tables, Aladin[4] for images, Splat[5] for spectra, and more.

Upcoming, there's ObsTAP that lets you post even more expressive queries against database tables.

These protocols also usually say how you can get the data once you have located it. There is work in progress on server-side manipulations, though (cutouts, cube cuts, etc) – but standards for that are hard.

## 14. Image Search in Aladin

(cf. Fig. 7)

This is a screen shot from Aladin 10, where I've discovered some historical plates from various image services. Note that on the sidebar on the left, the "resources" come from all kinds of different publishers. Aladin has just asked the Registry here.

## 15. Using Data

The VO uses existing data formats where they are appropriate (e.g., FITS for images). Where they aren't it uses its own: **VOTable**, containing rich metadata. This saves you from having to write code every time you want to use a new data source.

And it comes with descriptions, units, UCDs, and more. UCDs are a VO thing, too: spelled out, it's unified content descriptors. They are short strings that say what kind of physics a column represents: pos.eq.ra is a right ascension, phot.mag;em.opt.V is a visual magnitude, etc.

The VO also defines data models (e.g., for spectra) that say what metadata items are necessary for a useful description.

Clients: TOPCAT[6] and STILTS[7] for generic VOTables; the clients for typed interfaces also consume VOTables. See also Astropy and many other libraries

3 http://www.star.bris.ac.uk/~mbt/topcat/
4 http://aladin.u-strasbg.fr/aladin.gml
5 http://star-www.dur.ac.uk/~pdraper/splat/splat-vo/
6 http://www.star.bris.ac.uk/~mbt/topcat/
7 http://www.star.bristol.ac.uk/~mbt/stilts/



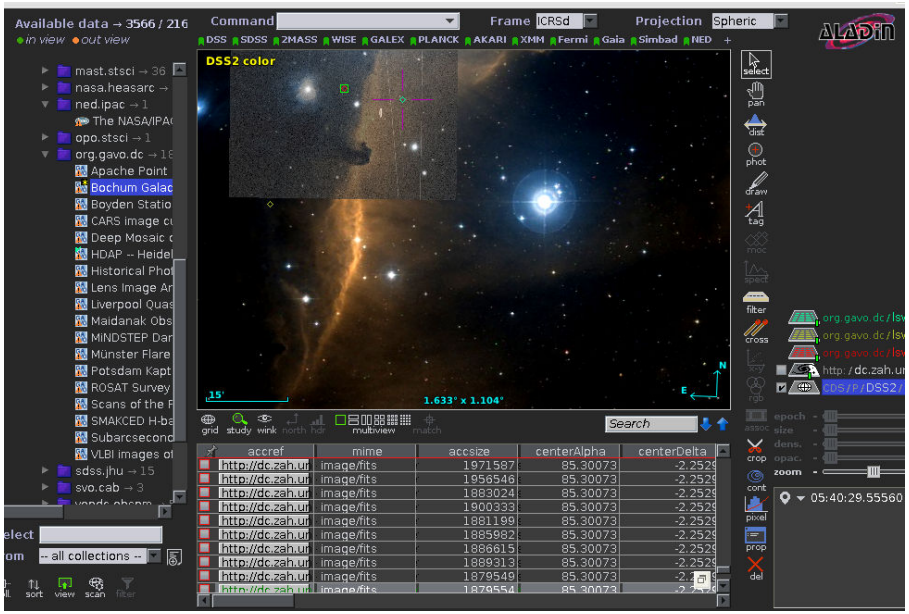


*Fig. 7*

## 16. Using Data Remotely

Some modern data collections are too large to move – smarts must come to the data.

**ADQL** lets you write simple programmes, **TAP** lets you run them on remote servers, upload your tables, and retrieve the results.

If you know CASJobs: about the same thing, only with a solid standard and supported by more services.

Clients: TOPCAT[8] and STILTS[9], TAPHandle[10], pyVO.

Learn it: ADQL course[11]

8 http://www.star.bris.ac.uk/~mbt/topcat/
9 http://www.star.bristol.ac.uk/~mbt/stilts/
10 http://saada.unistra.fr/taphandle
11 http://docs.g-vo.org/adql

# 17. From Tools to Toolkit

The VO is about standards. Any client implementing a standard can query any server implementing a standard. This gives users a choice of software, and using libraries or frameworks, they can simply write their own clients.

Plus, most VO software interoperates – you can send tables, selections, etc. from one programme to the next using a protocol called **SAMP**. Try it, it's fun.

Clients: Almost all of them. You won't even notice.

# 18. Demo time

Put together

- Registry
- TAP
- SSAP
- SAMP

to look for blue stars in the error circles of "hard" Antares sources.

(There's text explaining what I've done in the lecture notes) Here's what to do:

### Getting the Antares Data

Start TOPCAT, select VO/TAP. Type `antares` into the search box, select the GAVO service, hit "Run service".

TOPCAT then goes to the selected service and queries its table and column metadata, which you can presently browse. Use the search box near the top left corner to find your antares table.

You generally want to avoid downloading more data than is necessary. The collection here is small, so getting it all is cheap, but let's pretend we'd like to remotely figure out what "hard" (in the sense of: high-energy neutrinos) might mean for Antares data. In the table browser, you'll see that antares.data has a `n_hits` column. Let's make a histogram of it:

```
select round(n_hits/10)*10 as bin, count(*) as n
from antares.data
group by bin
```

(this is a piece of ADQL – and this type of query is about the fourth thing you'll learn in our ADQL course).

Look at the data in TOPCAT, perhaps making a plane plot. Based on this, I'd say "hard" would be $n_{\mathrm{hits}} > 100$.

Let's get those:

```
select *
from antares.data
where n_hits>100
```

(in general, things are going to be faster and smoother all around if you think first what columns you'll need and enumerate those rather than writing $*$; again, in Antares there aren't so many columns that not thinking first hurts).

Look at the initial data (e.g., a sky plot).



### Getting Blue Objects in the Error Disks

These days, the default for "give me point sources not far from optical" is of course Gaia. To see where you can query Gaia data, go back to the "Select Service" tab and type "Gaia". You can in principle use almost any service listed. I'll use the GAVO DC TAP again, where the Gaia DR2 table is called "gaia.dr2light". On other services, use the "Metadata" pane and its search box to find the respective tables.

What we want now is to crossmatch our local table with the Gaia table – TAP can do that using uploads. The query you need then is a bit complex, but TOPCAT can help you write it if you hit the "Examples" button (upload/upload join if you have your Antares subset selected in TOPCAT's table list and the Gaia table in the TAP window).

You'll have to edit it a bit until it does what you want: select the sensible columns, use the error radius as the match radius, add the colour constraint. In the end, this will look like:

```
SELECT
  source_id, ra, dec, phot_g_mean_mag,
  phot_bp_mean_mag-phot_rp_mean_mag as colour,
  n_hits, id
FROM gaia.dr2light AS db
  JOIN TAP_UPLOAD.t3 AS tc
  ON 1=CONTAINS(POINT('ICRS', db.ra, db.dec),
                CIRCLE('ICRS', tc.raj2000, tc.dej2000, ang_error))
  where phot_bp_mean_mag<phot_rp_mean_mag-0.5
```

(if the machine says something about not being able to write the request or somesuch, switch "Mode" to "Asynchronous").

Now draw a colour-magnitude diagram of the data, and perhaps a sky plot. Quite a few of the matches are probably spurious. But we might want to have a look anyway.

### Looking at things with Aladin

To deal with images in the VO, Aladin is a nice program. Start it, and tell TOPCAT to tell it where you're look using Views/Activation Actions. Check "Send Sky Coordinates".

In Aladin, activate the DSS layer and perhaps zoom in a bit. You'll see that the display follows the points you select in your CMD.

### Further investigation

To see if more is known about the objects that, after all, could be sources of your neutrinos, you could, in Aladin, activate the Simbad layer.

To look for a spectrum, in the first approximation: In the lower left corner of the Aladin window, there's a filter editor. In there, in the "Technical" Tab, check "SSA" (yeah, sometimes you're still seeing wires). Then, click on the SSA header in the resource tree, tell Aladin to only check for matches in the dialog popping up, and services that have data will turn green (if there are any).

When you get matches, you can send the matches back to TOPCAT (if you have image/fits spectra, you need a specialised client like Splat, though).

Well, for me, the matches I could make out were all white dwarfs. But there are VO ways to filter out the known ones, too. Ask me if you're curious.

# 19. Your Contribution

Do you have data that others could re-use? No? You're sure?

# 20. Common Excuses

Shamelessly stolen from http://datapub.cdlib.org/closed-data-excuses-excuses/[12]

- People will contact me to ask about stuff – well, science is about exchange, and you'll usually notice that most of those questions are actually quite clever, so answering them is a good use of your time.
- People will misinterpret the data – good documentation and standards mitigate this. The rest is just as with publishing prose, isn't it?
- My data is not very interesting – leave that decision to others. You'd be surprised how much „boring data" people click-and-type from printed graphs and tables each week.
- I might want to use it in a research paper – well, if you've not done so so far, will you? When? Too much data is gathering dust, waiting for the „real soon now". Be fair to the world and publish, if need be with an embargo.
- I'm not sure I own the data – that sucks. The original source has some advice for you.
- My data is too complicated – if it's too complicated to explain: are you sure you've understood it yourself? Try explaining anyway, you won't regret it.
- My data is embarrassingly bad – everyone's is. Good data is just bad data that more eyes have seen and more hands have improved.
- It's not a priority and I'm busy – ah-ha! Here we're coming to a real kicker. Rewarding data publishing is something we're working on (e.g., the Thomson Reuters has started a data citation index). Then again, publishing doesn't need to be so terribly painful...

# 21. Data Publishing

There is nothing like Journals for publishing data yet (though VizieR comes close for tables).

See: http://ivoa.net, "Publishing in the VO" – either:

- Ask a data center (VizieR, us, ...) to do it for you, or
- Use a publishing toolkit on your own machine, or
- Write your own software using libraries

# 22. Conclusion

The Virtual Observatory is there for you.

Use it.

Contribute.

Thanks!

[12] http://datapub.cdlib.org/closed-data-excuses-excuses/