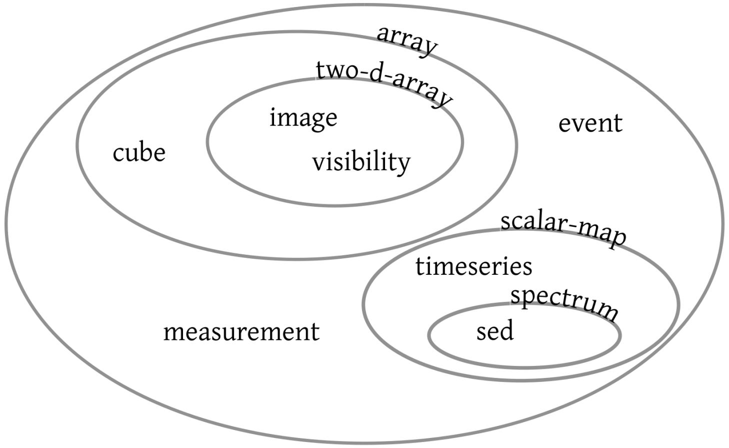

Fig. 1

Fig. 2

1. Product Type Revisted

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

Last interop: a tree with arrays, relational data, and dimensionality.

That was an immediate bummer.

That idea was questionable because, for instance, lots of spectra come in arrays. And something as plain as a dynamical spectrum couldn't sensibly be sorted in. Is it 2D data? Should there be a "vector-map" instead?

Back to the guiding use case.

(cf. Fig. 1)

(cf. Fig. 2)

2. Guiding Use Case

```
SELECT TOP 30 access_url
FROM ivoa.obscore
WHERE 1=gavo_vocmatch(
  'product-type',
  'spectrum',
  dataproduct_type)
```

should return everything that's reasonably spectrally resolved.

Plus, there's now content_qualifier in Datalink. We need product-type be ready for Datalink PR. The use case for that content qualifier, for all I can make out, is to let a datalink client figure out which client on a SAMP bus it might want to send a link to; there is semantics involved in that because the datalink client could try sending a dynamical spectrum to a client known to deal with spectra if it doesn't know one that knows about dynamical spectra in particular.

3. SKOS?

With these use cases, product types do not come out as a tree.

Example: a dynamical spectrum would be a child both of time series and of spectrum.

One could try to turn things around and make dynamical spectrum the parent concept with time series and spectrum as children. But then spectrum would reasonably have to a child of, say, space-spectrum-cube, too, and again we'd not have a tree.

Time to relax. Constraints.

4. SKOS.

The Simple Knowledge Organisation System has a very loose notion of "narrower". With that, there's nothing wrong with

#dynamical-spectrum	"narrower"	*#timeseries*
#dynamical-spectrum	"narrower"	*#spectrum*
**#spectral-cube*	"narrower"	*#image*
**#spectral-cube*	"narrower"	*#timeseries*

Downside: "narrower" is not transitive: our SKOS concepts can only look at their neighbours.

This is not just a whim. Yes, there are transitive variants of that narrower in SKOS, but given SKOS concepts form a graph rather than a tree (which, after all, is why we want them here), following anything in a SKOS vocabulary becomes a general graph traversal. Which is something you'd like to avoid as long as you can.

Hence, we should try hard to construct our vocabulary in a way that our use cases can be satisfied by clients just looking at the immediate neighbours.

5. Product type in SKOS

#image, *#cube*, *#spectrum*, *#timeseries*, *#visibility*, *#event*, *#measurements* from obscore are top-level.

#sed is narrower than *#spectrum*. This may seem odd, because a SED can very well consist of multiple spectra; but the choice becomes obvious once one asks "when a user looks for a spectrum, would they also want to find SEDs?"

#measurements still doesn't have a satisfactory definition. Help wanted!

With that:

#dynamical-spectrum is narrower than *#timeseries* and *#spectrum*. This would satisfy my search-for criterion: People looking for spectra probably will be happy to find dynamical spectra, even if it might take them a bit of work to figure them out.

That's not entirely trivial. If we modified our use case to: Let a client programme figure out what sort of data set it might be able to work with, then dynamical spectra should not be returned with spectra or time series. Our datalink use case is rather close to that, by the way.

6. More Terms

When people come forward who want them, we could add

- **#spectral-cube* (narrower than *#cube*, *#image*, and *#spectrum*)
- **#time-cube* (narrower than *#cube*, *#image*, and *#time-series*)
- **#slit-spectrum* (narrower than *#spectrum*)

Again, try the seach-for test: "people looking for an image will also want to find space-spectrum cubes".

Don't take my word for it, because I can't say I'm totally convinced that's "true". Here, "true" doesn't need to mean "*everyone* ever searching for images will be grateful for space-spectrum cubes". I maintain it should rather mean "averaged over our users, a possible annoyance because of data they can't immediately use is outweighed by the benefit of having at least some data."

This also lets me add a brief illustration of the transitivity problem: if we were to define a **#dynamical-sed*, we would have to explictly name *#spectrum* as wider of that. A client with SKOS could not infer that **#dynamical-sed* is narrower than *#spectrum* just because it is narrower than *#sed*; with RDF classes and properties, it could do that.

Is that a problem? Not in this case, certainly, if only because I don't think **#dynamical-sed* is a useful concept. I'm less sure that this won't fall on our feet in other cases.

7. Start Out?

As I said: we need to have something ready for the Datalink review.

Should it be this?

I'll write it up if nobody protests loudly here.
(You can still protest once you see what it is).

Thanks!