Fig. 1

Fig. 2

1. Axes Lengths Considered Harmful

(cf. Fig. 1)

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

(cf. Fig. 2)

Current proposal: New mandatory columns

s_dim1, s_dim2, t_tim, em_dim, pol_dim

each containing a pixel count.

- Why I'm frowning
- Alternatives

2. Reasons to frown

- 20% increase in number of columns (where the most common use case could be covered with just one column)
- With < 2.5 axes on average, more than 50% of the cells will be NULL (or 1?) – morbus denormalisitis
- Will require a schema change every time we want to support new data types

3. What about visibilities?

(we have a Radio priority, remember?)
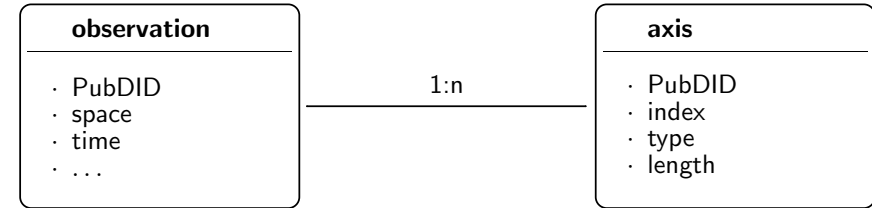
4. What about cartesian 3d?

(Gaia's merrily taking data...)

observation	1:n	axis
· PubDID		· PubDID
· space		· index
· time		· type
· ...		· length

Fig. 3

5. Alternatives

What we're modelling essentially is:

(cf. Fig. 3)

Normalized relational representation in `ivoa.obs_axes`:

PubDID	index	type	length
ivo://foo/bar?ex	0	pos	1200
ivo://foo/bar?ex	1	pos	1000
ivo://foo/bar?ex	2	spect	70
ivo://foo/bar?ex2	0	spect	700

6. Extra Table?

An extra table is not attractive. Alternatives

- Arrays – equivalently powerful, but needs support in backend and ADQL. Also, far less convenient query patterns
- Simulated array of column types (e.g., /pos/pos/spect/). Loses axis length, but has nice query patterns and is consistent with what we have in pol_states

7. Conclusions

Do we *absolutely* need the axis lengths?

[only use case known so far: detecting degenerated axes]

If not, the simulated column type array would be

- simpler (lower column count)
- more flexible (visibilities supported with one label we can probably take from FITS)
- consistent with existing obscore