1. Axes Lengths Considered Harmful

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

Current proposal: New mandatory columns
- \texttt{s\_dim1}, \texttt{s\_dim2}, \texttt{t\_tim}, \texttt{em\_dim}, \texttt{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 merely taking data...)

5. Alternatives

What we’re modelling essentially is:

What we’re modelling essentially is:

\texttt{ivoa.obsaxes}:

<table>
<thead>
<tr>
<th>PubDID</th>
<th>index</th>
<th>type</th>
<th>length</th>
</tr>
</thead>
<tbody>
<tr>
<td>/foo/bar/axes</td>
<td>0</td>
<td>pos</td>
<td>1200</td>
</tr>
<tr>
<td>/foo/bar/axes</td>
<td>1</td>
<td>pos</td>
<td>1000</td>
</tr>
<tr>
<td>/foo/bar/axes</td>
<td>2</td>
<td>spect</td>
<td>70</td>
</tr>
<tr>
<td>/foo/bar/axes*</td>
<td>0</td>
<td>spect</td>
<td>700</td>
</tr>
</tbody>
</table>

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 \texttt{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