Fig. 1

Fig. 2

Fig. 3

1. ArraySQL/SODA = SQL/TAP?

(cf. Fig. 1)

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

(cf. Fig. 2)

Premises:

- SODA will need complex operations on remote arrays
- Complex operations need an actual language
  (rather than or in addition to a bunch of parameters)
- TAP/ADQL are VO protocols that let people to complex operations on sets remotely
- Can we do a similar thing for arrays?

(cf. Fig. 3)

2. ISO 9075 Part 15

We're not the first with this use case.

The SQL committee (ISO 9075) as been developing SQL/MDA ("Array-SQL") for a while.

- Currently unpublished
- But 177 pages
- Main implementation rasdaman (U Bremen plus spin off)
- Works as an addon to several RDBMSes (commercially, there's even FITS support built-in)
- Current use mostly in Geosciences, apparently

3. The General Condense Operation

The SELECT statement of Array-SQL is:

```
condense condenserOp  // like +, min, max, card
over     var in mintervalExp // the indices
where    booleanExp   // as in SQL
using    cellExp      // what consenserOp works on
```

e.g., projecting a 3d array to 2d by summing up:

```
condense +
over x in sdom(x)[2]
values a[*:*, *:*, x[0]]
```

4. Pre-Defined Condenses

Many common operations are actually just Condenses, but are pre-defined for convenience, making things quite natural for numpy users.

Examples:

- Slicing: `mr[120:160, 55:75]`
- Section: `mr[*:*, sdom(mr)[1].hi, 0:256]`
- "Induction": `(4*mr+3*mv)/7`

5. Used in SQL

These operations are usually expected to sit within a SQL query, working on multiple arrays at once.

Example: retrieve 10-item histograms of all arrays in the mr table:

```
select marray v in [0:9]
       values condense +
       over x in sdom(mr)
       where mr[x]==v[0]
       using 1
from mr
```

This of course assumes that the domain of mr is 0 to 9.

That's not really convenient for SODA, where we assume we operate on basically a single image. We might encourage people to pass in multiple values for ID, though. Perhaps the strict distinction between discovery and analysis actually isn't all that necessary and useful.

6. But: WCS?

SQL/MDA has no idea of world coordinate systems.

Enter the *Web Coverage Processing Service* WCPS:

- Protocol-independent
- Heavily slanted towards geospatial stuff
- Discovery functions (dimensionList, imageCRS, imageCrsDomain...)
- Safe in evaluation (i.e., non-Turing complete)

Note: WCS*= OGC Web Coverage Service, roughly comparable to our SODA.

If you want to play around: There's a web-based form[1] with sample queries and all.

7. WCPS cutout

```
for b1 in (L8_B1_32631_30)
return
  encode(
    ( ( (b1*0.00002) - 0.1)
      [
        E(377983:390000),
        N(4902991:4917275),
        unix(1433068497)] ),
   "tiff")
```

As SQL/MDA, WCPS can operates on sets of arrays and can return more than one image (say) per query.

[1] https://eodataservice.org/wcps/

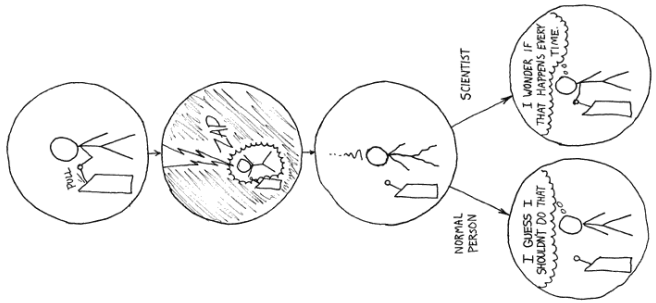

Fig. 4

8. Filtering with WCPS

The main trick for WCS apparently are the interval-generating functions ("E", "N", and "unix" in the example).

After that, you're back in integer land as in SQL/MDA. For instance, a Sobel filter:

```
coverage Sobel3x3
over
  px x ( -1 : 1 ),
  py y ( -1 : 1 )
value list
  < 1, 2, 1,
    0, 0, 0,
    -1, -2, -1 >
```

9. SODA's Future?

- SQL/MDA seems on a good way, but only one main implementation
- plain SQL/MDA doesn't help with WCS/metadata
- OGC WCPS looks almost complete (would need to define astro-specific "CRS")
- WCPS has wide client and server support
- But: Discovery and processing blur together compared to obscore+SODA
- Use TAP with LANG=WCPS?

10. Experiments, anyone?

(cf. Fig. 4)

Thanks