1. SSAP evolution 2012

Markus Demleitner (with input from Petr Skoda)
msdemlei@ari.uni-heidelberg.de

- The mythical `getData` operation: Cutouts and more
- Fuzzy querying for objects
- Object name discovery

2. getData: Why?

Primary usecase: Cutouts. For spectra, that's particularly useful since people frequently are interested in just the vicinity of a spectral line.

Secondary usecase: Generation of spectra in diverse formats. SSAP already has a `FORMAT` parameter; if a service supports generating lots of formats (e.g., VOTable, SDM FITS, "FITS image"), the SSAP response will contain a row each for every format and every object, which may be clunky depending on the smarts of the client; deferring the FORMAT decision to retrieval remedies that.

With cutouts, continuum subtraction, rebinning, and similar become difficult in the client. Thus, it may become necessary to put this on the server side, too.

3. getData: The principle

Idea: Rather than use what's returned as accref, use the publisher DID on `getData`.

`getData` without a PUBDID would return a metadata response.

Possible addition: Allow creator DID as well? This would potentially allow the discovery of a uniquely identified spectrum; however, since right now it seems nobody actually passes around creator DIDs, it's probably not worth the effort.

Unknown PUBDIDs yield a 404.

`getData` may evaluate further parameters, locally defined ones and parameters like `FORMAT` (on-the-fly format conversion), `BAND` (cutouts), `SPECRP` (rebinning), and `FLUXCALIB` (flux normalization).

`getData` must error out on unknown parameters or malformed parameter values.

4. PQL metacharacters

SSAP vaguely defines metacharacters for parameters: `7e-7/8e-7,1e-10/2e-10;RESTFRAME` might be a valid specification for `BAND`.

I propose to not recognize any of this and just define a special syntax for BAND: `[<low>]/[<high>]`.

Personal preference: I'd much rather have, say, `LAMBDA_LOW` and `LAMBDA_HIGH` and punt the parameter syntax entirely, but that's probably too far from SSAP.

Completely unhandled right now: transformation to rest frame. This may not be so bad for the primary use case of querying specific lines on specific objects since the object's redshift would usually be known to the client (or its user).

5. Generation Parameters

To declare support of `getData`, services include a TABLE named `generationParameters` in the SSAP `queryData` response (a copy of this table is returned when no PUBDID is given in `getData`):

```
<TABLE name="generationParameters">
  <PARAM name="FORMAT" datatype="char" arraysize="*"
    value="application/x-votable+xml">
    <VALUES>
      <OPTION>application/x-votable+xml</OPTION>
      <OPTION>text/plain</OPTION>
      <OPTION>application/fits</OPTION>
    </VALUES>
  </PARAM>
  <PARAM name="BAND" datatype="float" unit="m">
    <VALUES>
      <MIN>2e-7</MIN>
      <MAX>8e-7</MIN>
    </VALUES>
  </PARAM>
</TABLE>
```

Basically, there are enumerated parameters (probably all string-valued parameters are enumerated), and ranged parameters (probably all floating-point valued parameters). The VALUES children allow clients to figure out what values *might* yield a spectrum in from `getData` request. The service is still free to return a 400 with a standard SSAP error document if `getData` parameters are not palatable or come in an invalid combination.

6. Pattern Queries

For solar system objects, components of multiple star systems, exoplanets, etc., position-based queries are insufficient. TARGETNAME queries don't work well since literal matches are too restrictive given the quality of real metadata.

(Partial) mitigation: New SSAP parameters WILDTARGET and WILDTARGETCASE.

Servers supporting them declare so in their `FORMAT=Metadata` response.

Proposed pattern language: POSIX shell patterns (with *, ?, [], and backslash as metacharacters).

7. PQL Metacharacters, again

We probably want to allow specification of multiple patterns ("Beteigeuze,alf*ori,alp*ori"). Using PQL syntax with commas would require more pattern syntax, which I don't like.

Alternative: Require evaluation of multiple occurrences of WILDTARGET*.

Problem: This is against how SSAP currently works, and there's been resistance against repeated parameters.

8. Pattern language alternatives

- SQL patterns (% and _ as metacharacters)
- DOS patterns (* and ? as metacharacters with ad-hoc escaping)
- Some subset of Perl-compatible REs
- "Google-like"

There's a deliberation of these alternatives in the spec draft.

9. getTargetNames

Astronomical nomenclature is a mess. It would be useful if people could get a quick idea of what nomenclatures are in use on a given service.

Solution: `REQUEST=getTargetNames`. This is exactly equivalent to `queryData`, except only one row is returned for every distinct object name in the result set, and each row only contains one column with `utype="ssa:Target.Name"`.

Since `MAXREC` applies, even huge archives will not be swamped; overflows are signalled as in SSAP.

10. Implementation status

GAVO DaCHS implements `getTargetNames` – it's straightforward if you have an RDBMS behind your archive (who hasn't?)

GAVO DaCHS implements WILDTARGET* except for multiple parameter specs – the pattern translation is rather messy

No implementation exists for `getData`. If we actually go for stuff like server-side normalization, this will become ugly, but efforts to that effect are under way in Ondrejov.

Draft spec at http://docs.g-vo.org/ssaevolution.html[1].

[1] `http://docs.g-vo.org/ssaevolution.html`