1. SSAP Seen From A 2012 Client

Markus Demleitner, Margarida Castro Neves
gavo@ari.uni-heidelberg.de

- Harvesting SSAP server metadata
- The utype saga
- Input parameters and their story
- Excuses, excuses, and lame excuses

2. Harvesting Metadata

When passed `REQUEST=queryData` and `FORMAT=Metadata`, an SSAP server is supposed to return a VOTable containing input parameter and an empty output table, like:

```
<VOTABLE xmlns="http://www.ivoa.net/xml/VOTable/v1.2">
  <DESCRIPTION>TheoSSA provides spectral energy distributions [...]
  <INFO name="legal" value="When publishing research [...]
  <RESOURCE type="meta">
    <INFO name="SERVICE_PROTOCOL" value="1.04">SSAP</INFO>
    <PARAM name="INPUT:POS" datatype="char"
      utype="ssa:Char.SpatialAxis.Coverage.Location.Value" unit="deg,deg">
      <DESCRIPTION>ICRS position of target object</DESCRIPTION> [...]
    <PARAM datatype="double" ucd="phys.abund" name="INPUT:w_Si">
      <DESCRIPTION>Mass fraction of Si in the model computed.  [...]
    <TABLE name="results">
      <GROUP utype="stc:CatalogEntryLocation">
        <PARAM datatype="char" arraysize="*"
          utype="stc:AstroCoordSystem.SpaceFrame.CoordFlavor"
          value="SPHERICAL" name="CoordFlavor"/>
```

To see such a document in full, try:

```
curl -F REQUEST=queryData -FFORMAT=Metadata http://dc.g-vo.org/theossa/q/ssa/ssap.xml
```

3. The Sample

The Astrogrid registry had **47** SSAP services 2012-04-18.

4 servers caused connection problems (in particular non-200 HTTP status codes)

5 servers returned non-well-formed XML. Most of those were failed attempts at HTML instead of a VOTable.

2 servers returned an entirely empty VOTable.

6 servers returned a VOTable declaring a result table, but no input parameters.

30 servers returned one or more parameters, including **2** that were, laudibly, fixed after we had complained.

4. Metadata: impact assessment

Bad metadata responses are **really bad**.

Many services, in particular theory services, can only be sensibly used if the input parameters are known. Without a proper metadata response, there's no way to figure those out.

Returning random XML-like stuff is an additional trap for naive clients trying to parse them after W3C started to throttle requests for DTDs and such.

5. The utypes saga

All standard FIELDs in SSA VOTables must come with utypes with a data model name of ssa, and that data model name must be declared as an XML namespace. Disclaimer: I think the second part is a bad idea, so I'm not actually blaming services that don't do this properly.

Census 1Q 2011: Retrieve queryData responses and check how utypes are used. Some results:

- Service 1 declares sdm xmlns and has ssa: utypes
- Service 2 declares sed xmlns and has ssa: utypes
- Service 3 declares no xmlns and has only one FIELD with a utype, and that's an sdm: utype
- Quite a few services declare no xmlns and have utypes without a data model name
- Quite a few services declare no xmlns and have no utypes
- Quite a few services have no xmlns declaration but use proper ssa: utypes

6. Impact assessment: utypes

Missing or bad xmlns declarations are probably venial sins. They could spell trouble when significantly different SSA DM versions were out in the wild.

Wrong or missing data model names theoretically are evil, but it seems the industry standard for utype interpretation is to disregard the DM name, so it's not a disaster right now. It's annoying, though.

Having no utypes at all is **bad** since it forces clients to guess what columns are what.

7. Input parameters

Here's the most frequent input parameters found on querying the registred SSAP server's meta-data:

11	teff_min	min value for the effective temperature [...]
11	teff_max	max value for the effective temperature [...]
10	logg_min,logg_max	min/max value for Log(G) for the model
7	REQUEST	
5	meta_max,mata_min	min/max value for the metallicity
5	COMPRESS	Return compressed results?
4	z_min,z_max	comment is just min/max...
4	MTIME,MAXREC	...
3	WILDTARGET	Shell pattern of target observed

Remarkable (1 occurrence): TARGETSPECTYPE with description "The spectral type of the target. It is better to specify a range (like B2.5III/B6V..." '.

8. Impact assessment

1. Most services do not declare their standard input parameters
2. Most custom parameters come as max/min pairs

Is it bad?

(1) Yes, since servers must ignore parameters they don't know and then dump their whole data; thus clients need a reliable way to figure out when that will happen.

(2) Yes; not so much in itself but since it's unclear what parameters support PQL syntax and which don't, it's effectively impossible to come up with good user interfaces for client authors. Humans may of course be able to figure out what's meant, bit as the spectral ranges example shows, even that is sometimes a bit hard.

9. Lessons

- Validators must check input parameter presence
- *If* we want to keep PQL syntax, we need some way to communicate what parameters support what syntax
- Is it too late to switch to what people seem to like best: structured names? Structured names are parameter names that can be "parsed" and thus replace at least parts of the PQL syntax (say, _min, _max, maybe _step, maybe _pattern).
- Let's not define exotic features like namespace declarations for utypes long before there's a chance they'll actually be used. People will get the stuff wrong in so many ways that once we actually want to do something with these concepts, the bug variety is so large that it's all a huge minefield.