Fig. 1

Fig. 2

1. Introducing Datalink

(cf. Fig. 1)

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

(cf. Fig. 2)

- What is Datalink?
- Standalone Datalink
- Datalink in IVOA DAL protocols
- Server-side processing

Target audience: Implementors, data providers; most users still have to wait a bit.

2. Problem Domain

Modern datasets often consist of many distinct pieces, e.g.,

- Science data in different resolutions
- Masks for unusable regions
- Calibration data
- Different processing levels (e.g., merged Echelle spectra)
- Derived graphs
- Different formats
- Extracted sources
- Services for server-side processing (cutouts, rebinning...)

Datalink is a standard protocol to publish such datasets.

3. Simple Solution

For this, Datalink defines:

- A table schema and its VOTable serialisation
- A lightweight semantics for the classification of artefacts
- A service interface
- The application/x-votable;content=datalink media type

4. Table Schema

Datalink services return VOTables with main table having at least the following columns:

- ID – a dataset id assigned by the publisher. If your service is integrated into the VO, this should be a publisher dataset id, which is a special kind of URI. But the standard police won't come for you if you, for now, use something else. This id, anyway, is the same for all files belonging to the same dataset.
- access_url – where to get the file
- service_def – reference to a service def (later)
- error_message – something went wrong for this file
- description – Human-readable information on the file
- semantics – machine readable content declaration
- content_type – media type to expect
- content_length – size of the file in bytes

5. Sematics

The (mandatory) semantics column contains terms from a common vocabulary. Actually, what's in there is URLs, and there's RDF behind it, so one may hope clients will, at some point, be able to figure out that some user-defined term "is-a", say, calibration file. We're not nearly there yet, though.Terms include:

- #progenitor
- #derivation
- #auxiliary
  - #weight
  - #error
  - #noise
- #calibration
  - #bias
  - #dark
  - #flat
- ...

Vocab URI: http://www.ivoa.net/rdf/datalink

6. Service Interface

Datalink documents can be exchanged like normal files.

They are also the response of services with standard id

ivo://ivoa.net/std/DataLink#links-1.0

Interface to those: Trivially one parameter, ID.

(ID can be repeated, though, which is why it's a table column)

7. Examples

With a web browser, you can try

- http://dc.g-vo.org/kapteyn/q/web/form[1] – plate photos, wedge
- http://dc.g-vo.org/rosat/q/im/form[2] – different bands, background images...
- http://dc.zah.uni-heidelberg.de/flashheros/q/web/form[3] – datalinks in a datalink document (e.g., look for 24 Lib)

You can just submit empty forms, which will give you essentially random results.

That is fine here since all that matters here is the links in the datalink column. What you see in the browser is the result of a stylesheet; the document is a VOTable (try that by loading it up in TOPCAT). Note how the semantics is used to provide a preview; it's also used for coloring.

8. Datalink in DAL protocols

Now consider a Spectrum (SSAP) or Image (SIAP) service – there is only one access URL in the result table.

Choices:

- put a datalink document there – then legacy clients are out of luck
- put the main dataset file (say, FITS) there – then datalink-enabled clients have nothing to work on
- give one row each to both FITS and Datalink – then you have every dataset twice in the result display unless your client is smart

All of them suck but we can do better:

[1] http://dc.g-vo.org/kapteyn/q/web/form
[2] http://dc.g-vo.org/rosat/q/im/form
[3] http://dc.zah.uni-heidelberg.de/flashheros/q/web/form

9. Declaring Datalink support

In their DAL responses, services can declare there's a datalink service for the datasets returned:

```
<RESOURCE type="meta" utype="adhoc:service">
  <GROUP name="inputParams">
    <PARAM name="ID" ref="ssa_pubDID" value=""/>
  </GROUP>
  <PARAM name="standardID"
    value="ivo://ivoa.net/std/DataLink#links-1.0"/>
  <PARAM name="accessURL"
    value="http://localhost:8080/flashheros/q/sdl/dlmeta"/>
</RESOURCE>
```

In the wild: Example[4], near the bottom.

The main points:

- The declaration is in a VOTable resource with a utype of adhoc:service.
- There's a group defining input paramters; this is a datalink service, so it only has one, ID.
- The PARAM has a ref, which says: Take the value from that column; this is from an SSA service, which by the standard must come with a column containing the PubDID. That's referred here.
- Clients can tell by the standardID param that this is indeed datalink
- The accessURL param points to the base URL of the datalink service

So: Legacy clients get their FITSes, Datalink-enabled clients can trivially generate datalink URLs.

10. Server-Side Processing

Why not re-use service declarations for other purposes?

```
<RESOURCE ID="lwlltlid" type="meta" utype="adhoc:service">
  <GROUP name="inputParams">
    <PARAM name="FLUXCALIB" ucd="phot.calib" value="">
      <DESCRIPTION>Recalibrate the spectrum.  Right now, the only...
      <VALUES>
        <OPTION name="RELATIVE" value="RELATIVE"/>
        <OPTION name="NORMALIZED" value="NORMALIZED"/>
    <PARAM ID="lgntblid" datatype="float" name="LAMBDA_MIN"
      ucd="par.min;em.wl" unit="m" value="">
      <DESCRIPTION>Spectral cutout interval, lower limit</DESCRIPTION>
      <VALUES>
        <MIN value="3.4213e-07"/>
        <MAX value="8.6328e-07"/>
    ...
```

Important: Enough metadata for meaningful UIs. This pertains, in particular, to communicating the domains, units, and physics. An upcoming standard will give triples of name, UCD, and unit so clients can use canned UIs for parameters with common semantics ("lower limit in wavelength").

In the wild: Example[5], last but one RESOURCE.

[4] http://dc.g-vo.org/flashheros/q/ssa/ssap.xml?REQUEST=queryData
[5] http://dc.g-vo.org/flashheros/q/ssa/ssap.xml?REQUEST=queryData

11. #access rows

Service blocks are allowed in datalink responses, too; references to them are in the service_def column of the datalink response. From a Flash/Heros spectrum[6]:

```
   <TR>
pubDID      <TD>ivo://org.gavo.dc/~?flashheros/data/ca92/f0011.mt</TD>
service_def <TD>lgltoued</TD>
semantics   <TD>#access</TD>
...
  <RESOURCE ID="lgltoued" type="meta" utype="adhoc:service">
    <GROUP name="inputParams">
      <PARAM name="FLUXCALIB" ucd="phot.calib" value="">
        <DESCRIPTION>Recalibrate the spectrum.  Right now, ...
        <VALUES>
          <OPTION name="RELATIVE" value="RELATIVE"/>
    ....
```

Splat can already use this. Demo?

12. Conclusion

Got complex datasets? Use datalink documents to describe them!

- Future clients will know what to do
- Until then, XSLT lets you produce web UIs

Maintaining a client? Teach it Datalink!

- Basically, easy to understand
- Lightweight semantics enables attractive UIs with low effort

Just a user? Ask data providers and client writers for Datalink!

- Uniform, file-exact access
- Server-side data manipulation for great savings in transfer volume.

13. Lecture Notes

http://docs.g-vo.org/talks/2015-kiel-dl.pdf[7]

This link won't last forever; if you read this after Sept 2015, you probably know where you got it from.

[6] http://dc.g-vo.org/flashheros/q/sdl/dlmeta?ID=ivo://org.gavo.dc/~?flashheros/data/ca92/f0011.mt

[7] http://docs.g-vo.org/talks/2015-kiel-dl.pdf