Fig. 1

Fig. 2

1. Datalink your Complex Dataset

(cf. Fig. 1)

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

(cf. Fig. 2)

- What is datalink?
- Deployment scenarios
- A closer look at the schema
- Also: Services

2. What is datalink?

Modern datasets rarely are flat files. Instead there is:

- Science data in different resolutions
- Masks for unusable regions
- Calibration data
- Extracted sources
- Rich provenance (e.g., source images for stacking)
- . . .
- Services for server-side processing (cutouts, rebinning. . .)

Datalink lets you package all this up in an interoperable way, and without forcing users to pull Gigabytes of tar files.

3. Ingredients

The standard defines:

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

4. Links to data

Datalinks are described in tables with 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

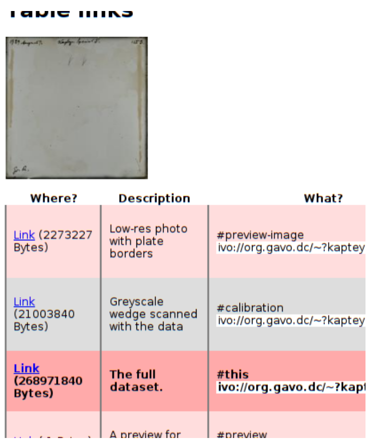
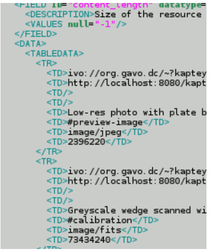


Table links

Where?	Description	What?
Link (2273227 Bytes)	Low-res photo with plate borders	#preview-image ivo://org.gavo.dc/~?kaptey
Link (21003840 Bytes)	Greyscale wedge scanned with the data	#calibration ivo://org.gavo.dc/~?kaptey
Link (268971840 Bytes)	**The full dataset.**	**#this** **ivo://org.gavo.dc/~?kapt**
	A preview for	#preview

Fig. 3



```
<FIELD ID="content_length" datatype=
  <DESCRIPTION>Size of the resource
  <VALUES null="-1"/>
</FIELD>
<DATA>
  <TABLEDATA>
    <TR>
      <TD>ivo://org.gavo.dc/~?kaptey
      <TD>http://localhost:8080/kapt
      <TD/>
      <TD/>
      <TD>Low-res photo with plate b
      <TD>#preview-image</TD>
      <TD>image/jpeg</TD>
      <TD>2396220</TD>
    </TR>
    <TR>
      <TD>ivo://org.gavo.dc/~?kaptey
      <TD>http://localhost:8080/kapt
      <TD/>
      <TD/>
      <TD>Greyscale wedge scanned wi
      <TD>#calibration</TD>
      <TD>image/fits</TD>
      <TD>73434240</TD>
```

Fig. 4

5. An Example

(cf. Fig. 3)

This is a plate scan with the actual data, a simple preview, a low-res photograph, and a greyscale wedge as an example for calibration data. This comes with file sizes, human-readable description, and hints for computers.

6. Not Just a Web Page

The previous screen shot is not just a web page, it is machine-readable data. (We used a publicly available style sheet[1] to display it in the browser, on which you're welcome to collaborate)

(cf. Fig. 4)

[1] https://github.com/msdemlei/datalink-xslt.git

7. Deployment Scenarios

- Standalone
- Within DAL services (SIAP, SSA, Obscore. . .)

Datalink documents can be shipped out like other data products; use the application/x-votable;content=datalink media type

Recursive datalink: datalink documents can be targets of datalinks – e.g., related datasets, drill down into provenance, etc.

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:

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[2], 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.

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

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[3], last but one RESOURCE.

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[4]:

```
   <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. Conclusions

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

Your users will benefit:

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

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

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