1. Linked Data in VOTables?

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

- What is Linked Data?
- Reminder: RDF
- Reminder: RDFa in TAP examples
- Trivial RDFa lite in VOTable metadata
- Less Trivial RDFa lite in VOTable metadata
- RDFa lite in VOTable TABLEDATA
- Should we go there?

Funded by e-inf-astro, BMBF FKZ 05A20VH5

2. What is Linked Data?

In current practice: Just another buzzword.

Potentially: Rich, machine-readable metadata for otherwise "normal" data files using embedded annotation understandable to *standard* (non-VO) tools...

...linking data to metadata items, complex entities, and possibly other data using RDF.

3. Remind me: RDF

RDF, the Resource Description Framework, represents information about "resources" (really: anything that has an URI) in triples of

$$\langle Subject, Predicate, Object \rangle.$$

All of these usually are URIs. Think of datalink: a datalink row could be (partially) represented in the RDF triple:

⟨ivo://org.gavo.dc/ ?plts/data/1960Sep_PLS/04573_1.fits,
http://ivoa.net/rdf/datalink/core#documentation,
http://dc.g-vo.org/plts/q/dl/static/marked/4573.jpg⟩

Which means: The thing with the publisher DID ivo://...(a plate scan) has documentation (that URL points to a term we have defined in an IVOA vocabulary, so clients can figure out its definition and come up with a nice label for it) on the jpeg at http://dc.g-vo.org/...

You can build very complex data structures in this way – and then make computers reason about them if you are courageous. For those old enough to remember the Semantic Web hype: This is what it was about.

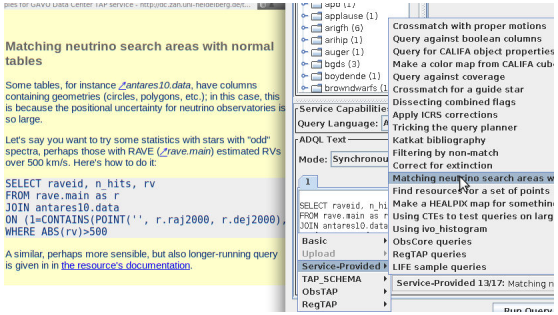

Fig. 1

4. Remind me: RDFa

There are many ways to write such triples. Most of them are fairly scary. A rather nice one in RDFa, which we are already using in DALI examples, which lets TOPCAT extract TAP example queries from things rendering just fine as web pages:

(cf. Fig. 1)

5. RDFa Source Code

There's actually not much magic behind this. Look at the source[1]:

```
<body vocab="http://www.ivoa.net/rdf/examples#">
  <div typeof="example" id="Matchingneutrinosearchareaswithnormaltables"
   resource="#Matchingneutrinosearchareaswithnormaltables">
    <h2 property="name">Matching neutrino search areas with
      normal tables</h2>
    <p>Some tables, ...</p>
    <pre class="dachs-ex-tapquery literal-block" property="query">
SELECT raveid, n_hits, rv
FROM rave.main as r
JOIN antares10.data
ON (1=CONTAINS(POINT('', r.raj2000, r.dej2000), origin_est))
WHERE ABS(rv)&gt;500
    </pre>
  </div>
</body>
```

Appreciate for a second what is going on here: This is normal HTML with a few extra attributes sprinkled in:

- with @vocab (here in body), I am defining a URI fragment that is prepended to any RDF resource references in within the element.
- With @typeof, I'm saying I'm introducing a resource of a certain RDF type (which is little more than a statement that that resource is eligible as subject or object of certain properties). Here, we are saying: "this div has the type http://www.ivoa.net/ref/examples#example" (I'll leave out the vocab prefix from here on).
- I'm giving that div an id and then, by referencing it in @resource, tell RDF engines that's the subject in triples to come.

[1] `http://dc.g-vo.org/tap/examples`

- Via @property, I am giving a predicate of an RDF triple. The subject is what I've just set via @resource (or the whole document by default), the object is the element content (or perhaps other things). That works both for the name property for the h2 element and for the query property in the pre element.

6. For VOTable?

Minimal RDFa ("RDFa lite") only needs the attributes vocab, typeof, property, resource, prefix, and possibly href and src.

If we allowed these for some (GROUP, INFO, PARAM) or all (TR, TD!) VOTable elements, we could make certain VOTable metadata readable for non-VO software.

7. Level 0: Atomic Dublin Core

Quite a bit of our VOTable metadata maps nicely to Dublin Core. Standard tools can interpret them when we write:

```
<VOTABLE...>
  <DESCRIPTION property="dcterms:description">
    This schema contains data re-published from the official...
  </DESCRIPTION>
  <RESOURCE type="results">
  <INFO property="dc:rights" name="copyright">
    If you use public Gaia DR3 data in a paper, please take note of
    ...
  </INFO>
```


8. Level 0: It Works!

I've written a little program on top of pyrdfa, somewhat convolved because the main library entry point is a bit funky at the moment:

```
proc = pyRdfa()
proc.media_type = "text/html"  # that's cheating
# source detection currently is horribly broken in pyrdfa
print(proc.rdf_from_source(io.StringIO(open(sys.argv[1]).read())))
```

This prints the RDF triples extracted from the document passed on the command line (turtle notation):

```
$ get-triples.py withdc.py
@prefix dc: <http://purl.org/dc/terms/> .

<>
  dc:description """
    This schema contains data re-published from the official
    ..."""" ;
  dc:rights """If you use public Gaia DR3 data in a paper, please...
    """ .
```

This means: We have a description and a legal notice on the current document (that's what <> means in turtle), and here's what they read. Experts beware: I've used the dc: prefix here without binding it to any URI because RDFa lite predefines it; as you can see, it correctly shows up in the output turtle.

9. Level 1: Typed Annotations

There's a ready-made vocabulary "description of a project" (doap) that states how, well, projects are described. We could declare the project that created the data like this:

```
<GROUP property="dc:creator"
  id="srcproj"
  resource="#srcproj"
  typeof="doap:Project"
  prefix="doap: http://usefulinc.com/ns/doap#">
        <INFO property="doap:name">DPAC consortium</INFO>
        <INFO property="doap:homepage"
          >https://www.cosmos.esa.int/web/gaia/dpac/consortium</INFO>
</GROUP>
```

Note that RDFa does not know the doap prefix by default, and hence we have to declare it in a prefix attribute.

10. Level 1: It Works!

Again, standard RDFa tooling can pull out structured information:

```
$ get-triples.py with-typed-ann.xml
@prefix dc: <http://purl.org/dc/terms/> .
@prefix doap: <http://usefulinc.com/ns/doap#> .

<> dc:creator <#srcproj> .

<#srcproj> a doap:Project ;
    doap:homepage "https://www.cosmos.esa.int/web/gaia/dpac/consortium" ;
    doap:name "DPAC consortium" .
```

So: the Dublin Core creator of the VOTable is a project, and that project has a name and a homepage (and possibly much more doap metadata).

Note that this is not quite right because the homepage is a literal (i.e., just an opaque string, as evinced by the double quotes) rather than a resource (which would have angle brackets). Within RDFa, the natural way of getting things to be resources would be to have href attributes to mark up resource references:

```
<INFO property="doap:homepage"
  href="https://www.cosmos.esa.int/web/gaia/dpac/consortium"
  >https://www.cosmos.esa.int/web/gaia/dpac/consortium</INFO>
```

– as an alternative to @value, that may be be an option that's actually in the RDFa spirit. Even outside of resource references, @value would probably be what RDFa engines should pick up from an INFO element rather than the element content, but regrettably there's no way to tell them that.

It is conceivable that using full RDFa and adding a `datatype='rdfs:Resource'` could fix this problem while keeping the URI in the content; I've not thought about it deeply enough but note that with current pyrdfa, it doesn't come out right.

11. Level 2: Datalink By The Book

Datalink already is RDF-enabled via semantics. We *could* expose the *data content* as RDF, too:

```
<FIELD arraysize="*" datatype="char" name="ID"/>
<FIELD arraysize="*" datatype="char" name="access_url"/>
<FIELD arraysize="*" datatype="char" name="semantics"/>

<TABLEDATA vocab="http://www.ivoa.net/rdf/datalink/core#"
  resource="ivo://org.gavo.dc/~?kapteyn/data/fits/POT015_000317.fits">
<TR>
  <TD>ivo://org.gavo.dc/~?kapteyn/data/fits/POT015_000317.fits</TD>
  <TD property="preview-image" >http://dc.g-vo.org/[...]/POT015_000317.jpg</TD>
  <TD>#preview-image</TD>
</TR>
```


12. Level 2: It Works!

I've added two more rows like this; the VOTable can then be digested to:

```
@prefix ns1: <http://www.ivoa.net/rdf/datalink/core#> .
@prefix ns2: <http://www.w3.org/ns/rdfa#> .

<> ns2:usesVocabulary ns1: .

<ivo://org.gavo.dc/~?kapteyn/data/fits/POT015_000317.fits>
  ns1:calibration
  "http://dc.g-vo.org/kapteyn/q/dl/static/wedges/POT015_000317w.fits" ;
  ns1:preview
  "http://dc.g-vo.org/kapteyn/q/dl/static/jpegs/thumb_POT015_000317.jpg" ;
  ns1:preview-image
  "http://dc.g-vo.org/kapteyn/q/dl/static/jpegs/POT015_000317.jpg" ;
  ns1:this
  "http://dc.g-vo.org/getproduct/kapteyn/data/fits/POT015_000317.fits" .
```

Here, the type problem (with literals instead of resources) is even more dramatic, and a moderate amount of experimentation has not turned up a way to get pyrdfa to churn out the right turtle. It feels as if that's my boneheadness. So: I'm not saying that's necessarily a showstopper, but we'd have to reach out to actual RDFa experts or otherwise wait for enlightenment.

Also note that in more typical settings the resource attribute would typically sit on the TR and would probably do the id/resource trick we did for the doap GROUP above. Datalink documents as God wanted them, however, only talk about one dataset (yes: multi-ID still considered harmful).

13. Should we?

- Is there sufficient external takeup of RDFa to make it worthwhile?
- Do we even want non-Astronomers to understand our VOTables?
- Would RDF tooling help us do our job? Note all the great vocabularies[2] other people have already come up with...
- Major ouch factors: VOTable ID vs RDFa id, links in literals or non-href attributes, no data annotation in BINARY/FITS serialised tables.

...Opinions?

[2] https://lov.linkeddata.es/dataset/lov/vocabs