



Fig. 1

1. Vocabularies 2: First WD

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

- Reminder: Goals, Non-Goals
- Three flavours
- Vocabulary maintenance
- Deployment issues
- revovo
- Uncertainty and Doubt

(cf. Fig. 1)

2. Vocabularies 2: Goals

- Standardise “datalink-like” vocabularies – of which there are by now quite a few more, from relationship types to reference frames; so far, only SKOS has been (somewhat) standardised
- Have clear rules for how terms are added (or modified)
- Have clear rules for how to consume the data without fat RDF tooling

3. Vocabularies 2: Non-Goal

RDF-ising the VO.

There would be something to be said for adopting RDFa, having some piece of software that maps VO-DML into RDF ontologies, and then treating VOTable as just another (rather compact) format to store (amazing amounts of) RDF triples. And perhaps that’s a good way out of the annotation crisis we’re having in DM for a long, long while.

But Vocabularies in the VO is *not* the standard to do it in.

4. Three Flavours

Perhaps the craziest thing about Voc2: three vocabulary flavours.

Flavour	Terms	Super
SKOS	skos:Concept	skos:broader
RDF property	rdf:Property	rdfs:subPropertyOf
RDF class	rdfs:Class	rdfs:subClassOf

RDF class and property vocabularies are based on RDFS and also use its properties for metadata (rdfs:comment and rdfs:label), whereas SKOS retains its skos:prefLabel and skos:definition.

What madness?

5. Yet there is method in’t

- SKOS: Lexicographic, “soft” hierarchy (e.g., “bicycle” can very well be wider than “handle bar”).
- RDF property: terms that link things (e.g., is-preview-for) but is a strict is-a hierarchy (e.g., a flatfield is calibration data).
- RDF class: terms that don’t (naturally) link things (e.g., ICRS). Again, strict hierarchy.

Admittedly: if we started from scratch, we’d build it differently. You might ask: then why don’t we start from scratch? There’s not much in this way going on in the VO right now. And there’s not so terribly much I could say to refute that. See Uncertainty and Doubt below.

6. Vocabulary Maintenance

Voc2 strives for minimal initial sets of terms, growing them using VEPs:

Vocabulary: <http://www.ivoa.net/rdf/datalink/core>
 Author: msdemlei@ari.uni-heidelberg.de
 Date: 2019-07-19
 New Term: IsPreviousVersionOf
 Action: Addition
 Label: Newer Version
 Description: This dataset in a previous edition, e.g., processed with an older pipeline, as part of an older data release.
 Relationships: rdfs:subPropertyOf(this)

Rationale:

The term is mainly intended for projects with data releases. IsPreviousVersionOf allows services to mark up links to (typically datalink documents for) later version(s) of this data set. It allows a client to alert users that a newer, probably improved, rendition of the current dataset is available and should presumably be used instead of what they are looking at.

7. Terms Used in Maintenance

An important mechanism is that once a VEP is there, the terms proposed become second-class citizens of the vocabularies, so people can orderly try things out; I also hope this is going to foster discussion. To make that happen, there's a special, IVOA-specific property. And there's two more administrative properties.

- `ivoasem:preliminary` – this term is under review (i.e., was added in response to an open VEP)
- `ivoasem:deprecated` – don't use this term any more
- `ivoasem:useInstead` – if a deprecated term has successor(s), they're the objects of this property.

I'd say these are bound closely enough to IVOA processes that we ought to define them ourselves.

8. Deployment Issues

- We're stuffing everything (directly) below <http://www.ivoa.net/rdf>.
- About 30% of the crazy stuff in Voc2 is a direct consequence of: You get RDF/XML out of the repo if you want, and you have clear rules for how to parse this with good ol' Xpath.
- Alternatives: HTML (default), TTL (most readable). Could grow that list.

9. revovo

The current draft comes with a Python module to try things:

```
import revovo
```

```

voc = revovo.load_vocabulary(
    "http://www.ivoa.net/rdf/refframe")
for term, (label, desc) in voc.terms.items():
    tpl = '#{17}', "{:25}": {}'
    if term in voc.deprecated_terms:
        tpl = '('+tpl+')'

    print(tpl.format(term, label, desc[:25]))
```

Try it! You'll have to pull <https://volute.g-vo.org/svn/trunk/projects/semantics/Vocabularies/revovo.py> into the same directory as this script (in the simplest case).

10. Uncertainty and Doubt

- Should we do RDF at all? You see, for now it seems to me that RDF tooling isn't so terribly great that it's worth going into a lot of trouble for interoperability. And we *are* going into quite a bit of trouble over, say, an ad-hoc some tree model in JSON.
- Should we ask people to parse turtle? If you want to *understand* what's going on (rather than just follow recipes), that's so much simpler in turtle than in the admittedly slightly insane RDF/XML.
- Do I worry too much about class vs. property vs. concept? Will anyone ever care?
- Is the VEP process too heavyweight? Too lightweight? Does the TCG even need to be involved? Re-use the UCD committee instead?
- The UAT (SKOS) is only mentioned in passing. Should we IVOA-ize it and offer it with our guarantees and with an `ivoa` URI instead?