STC in VOTable

Markus Demleitner *(msdemlei@ari.uni-heidelberg.de)*

Current state: Ochsenbein, McDowell, Rots, IVOA Note, 2009-06-12: Serialization using utypes; system definition in a group with utype AstroCoordSystem, column binding in a group with utype AstroCoords.

This talk: some suggestions for streamlining, RFC.

And, by the way, the STC library I would otherwise have talked about – see http://vo.ari.uni-heidelberg.de/soft

The AstroCoordSystem group

```
<GROUP utype="stc:AstroCoordSystem" ID="HIP" >
  <GROUP utype="stc:AstroCoordSystem.TimeFrame" >
    <PARAM name="TimeScale" datatype="char" arraysize="*"
      utype="stc:AstroCoordSystem.TimeFrame.TimeScale"
      value="TT" />
    [...]
  </GROUP>
  <GROUP utype="stc:AstroCoordSystem.SpaceFrame" >
    <PARAM name="CooFrame" datatype="char" arraysize="*"
     utype="stc:AstroCoordSystem.SpaceFrame.CoordRefFrame"
     value="ICRS" />
  [...]
```

– the utype/value pairs can be inferred from STC-X with a little effort. Additional hierarchy by grouping utypes belonging to the various frames..

The AstroCoord group

```
<GROUP ID="HIPcoo" utype="stc:AstroCoords" ref="HIP">
  <PARAM name="Jepoch" datatype="double" unit="yr"
    utype="stc:AstroCoords.Position.Epoch"
    value="1991.25" />
  <PARAM name="epochScale" datatype="char" value="J"
    utype="stc:AstroCoords.Position.Epoch.Scale" />
  <FIELDref ref="pm2"/>
  [...]
</GROUP>
```

– additional key-value pairs, a ref to the coordinate system, and id for column referencing and, optionally, references to the fields.

FIELD definitions

```
<FIELD name="RA(ICRS)" ucd="pos.eq.ra;meta.main"
  ref="HIPcoo" ID="RA1" datatype="double" unit="deg"
  utype="stc:AstroCoords.Position2D.Value2.C1">
</FIELD>
```

– utypes directly inferrable from an STC-X representation, reference to the AstroCoord group.

Change 1: Reverse References

Instead of having utype and ref on FIELD, put groups into the AstroCoords group:

```
<GROUP ID="lltoush_coo" ref="lltoush"
    utype="stc:AstroCoords">
  <GROUP ref="alpha"
    utype="stc:AstroCoords.Position2D.Value2.C1" />
  <GROUP ref="rv"
    utype="stc:AstroCoords.Redshift.Value" />
 </GROUP>
```

- Keep STC information confined to STC groups (helps libraries)
- Don't clobber utype and ref on FIELDs to preserve them for other, less generic purposes

Change 2: Flat systems

Just have all utype/value params as direct children of the AstroCoordSystem group:

```
<GROUP ID="lltoush" utype="stc:AstroCoordSystem">
  <PARAM arraysize="*" datatype="char" value="VELOCITY"
    utype="stc:AstroCoordSystem.RedshiftFrame.value_type" />
  <PARAM arraysize="*" datatype="char" value="ICRS"
    utype="stc:AstroCoordSystem.SpaceFrame.CoordRefFrame" />
</GROUP>
```

- Flat is better than nested (try `python -c ’import this’`).
- Additional groups add no information, probably don’t really help implementations or humans on parsing and complicate writing.

Change 3: stc: means package

Don't pretend the stc: in the utype has anything to do with an XML namespace.

```
<VOTABLE version="1.2" xmlns:xsi="http://www.w3.org/2
 xmlns="http://www.ivoa.net/xml/VOTable/v1.2">
```

- No xmlns:stc attribute here to avoid misconceptions
- reinforce the notion that the value of the utype attribute is an opaque string (from a machine's point of view)

Change 4: Only allow string values

Define that all STC PARAMs are `datatype= "char" arraysize="*"`.

- We have no real serialization rules for anything but strings for the PARAM's value attribute, do we?
- Provides the easiest way to unambiguously define the utype serialization by pointing to the STC-X schema
- Otherwise, libraries have to keep a mapping from “known utypes“ to their types; possible, but not nice.

What next?

- Tell me what you think.
- With sufficient encouragement, I'll try to prepare a new version of the Note.
- With even more encouragement, I might write an XSLT that could formally define the whole scheme based on STC-X
- I can quickly whip up an implementation of this scheme, sans any changes you detest.
- At least the reference reversal should definitely be done, or we'll suffer forever by effectively invalidating FIELD's utypes for domain data models.