UTypes discussion

DM 2

wednesday Oct 29th a-m

Reminder: What is a Utype

- A Utype is a tag ; it works as a name used in the VO
  - It says what is the name used in a specific data model to represent a concept:

    Example : in a FITS spectrum file where is the *spectral resolution* along the wavelength axis for example

    SPECRES → **Spectrum.Char.SpectralAxis.Resolution** is the adequate Utype in SpectrumDM.

    PIXSIZE → **SpatialAxis.SamplingPrecision.SamplingPeriod**

- It also says which type is used to represent this concept: simple value, structured data type, i.e a class.
  - A central position refers to a coordinate system and 2 coordinates (Image case)
- It is built up as a pointer inside the XML schema of the model
- gives access to the full data structure :

Starting from an XML Example

XML serialisation

```
<characterisationAxis>
<axisName>spatial</axisName>
<ucd>pos</ucd>
<unit>deg</unit>
<coordsystem id="TT-ICRS-TOPO" xlink:type="simple«
xlink:href="ivo://STClib/CoordSys#TT-ICRS-TOPO"/>
<coverage>
<location>
     <coord coord_system_id="TT-ICRS-TOPO">
          <stc:Position2D>
            <stc:Name1>RA</stc:Name1>
            <stc:Name2>Dec</stc:Name2>
           <stc:Value2>
                       <stc:C1>132.4210</stc:C1>
                       <stc:C2>12.1232</stc:C2>
           </stc:Value2>
          </stc:Position2D>
     </coord>
</location>   …
</coverage>
</characterisationAxis>
```

cha:characterisationAxis.Coordsystem

cha:characterisationAxis.Coverage.location

cha:characterisationAxis.Coverage.location
? **stc:Position2D.Value2D.C1**

How to build up the pointer

- From a UML Model (graph), derive an XML schema (tree) as above
  - Each node of the tree can be a Utype.
    - Lower level nodes (leaves) correspond to attributes of classes in the model (s)
    - Intermediate nodes correspond to classes , possibly embedded in other classes
- XML Serialisation can be
  - « hand carved » : Characterisation and Spectrum DMs
  - Automated
    - Cf Theory IG effort for rationalising this process

      UML →XMI → XSD → Utype list

Requirements on the Data Model design

- Few levels of intrication between classes (denormalisation)
- Avoid parametrised classes:

```
token1.token2[attribute2=valueA].token3.token4[attribute4=valueB]
```

- Inheritance / Substitution

  ex CharAxis.ndim --> ndim1
  --> ndim2 2D coords
  --> ndim3 3D coords

  - Do we allow all possible substitutions if various representations are needed :
    - spatial resolution in 2D can be a radius, an ellipsoid, a couple (size1, size2)
  - lowest level of representation is needed

Different views on Utypes

- For a scientific use (new comer to the VO)
  - How is a metadata concept (f. i. *resolution*) expressed in various models (STC, Spectrum, etc.)
  - not a uniform vocabulary:depends on the model
- From models Utypes represent
  - A **flat serialisation** that gives all the pieces of a data model : classes and attributes names
  - Can express the **hierarchy** between classes and attributes
  - Useful in Votable with groups – F.Ochsenbein
  - Is redundant because all XML nodes may have Utypes

Various uses of Utypes :data publishing

- Publish (new) data to the VO
  - At the user level
    - How is the bounding box of an observation represented?
    - How can I describe a physical axis in the VO?
    - Interactive tools : CAMEA , an editor for Characterisation serialisation (CDS/VOTech)
  - At the data collection level

Keyword mapping

FITS to dal interface or data model Utypes:

MEX	(ESO)	dal interface
DM_Mapper	(ESA)	dal interface
Interactive mapping tool	(CDS)	Characterisation/all DMs

Various uses of Utypes :DAL protocols

- Protocols
  - ex: SSA Rec :
    - « A UTYPE is a fixed string which uniquely identifies a field of a data model irregardless of representation. »
  - ex: SSA or Spectrum element

Observation file name = http://sdss.jhu.edu/images/sdss/10314.fits

Utype → **ssa:Access.Reference**

  - The protocols generally use utypes pointing to leaves of a Datamodel:
    - single values or simple compositions like Coos in arrays

Various uses of Utypes :database queries

Various prototype implementations :

- ASPID-SR (Igor Chilingarian) on Char

```
SELECT * FROM processed_data        WHERE
''cha:SpatialAxis.coverage.location.coord::stc:position2d.value2
.c1' <=308.798821                                     AND
cha:SpatialAxis.coverage.location.coord::stc:position2d.value2.c
1' >=308.512238                                       AND
'cha:SpatialAxis.coverage.location.coord::stc:position2d.value2.
c2' <= +60.353806                                     AND
'cha:SpatialAxis.coverage.location.coord::stc:position2d.value2.
c2' >=+60.069312                                      AND
'cha:SpatialAxis.numbins2.i1' >1      AND
'cha:SpatialAxis.numbins2.i2' >1      AND
'cha:SpectralAxis.numbins1'>1         AND
'cha:SpectralAxis.coverage.location.coord::stc:spectral.value
<2.02e-06
;
```

- SAADA (Laurent Michel, CDS) on Char DM
- DM_Mapper (A. Stebe) at ESAC →presentation

Various uses of Utypes :Applications

- Single values

  Ex in a deconvolution program for images:

  - PSF FWHM → cha:SpatialAxis.Resolution.ReferenceValue.RefVal

- Classes :

  Coordinates transformations, resampling

  - Central Coordinate → cha:SpatialAxis.coverage.Location

```
<cha:location>
  <cha:coord coord_system_id="TT-ICRS-TOPO">
    <stc:Position2D>
      <stc:Name1>RA</stc:Name1>
      <stc:Name2>Dec</stc:Name2>
      <stc:Value2>
        <stc:C1>132.4210</stc:C1>
        <stc:C2>12.1232</stc:C2>
      </stc:Value2>
    </stc:Position2D>
  </cha:coord>
</cha:location>
```

  - Region description cha:SpatialAxis.coverage.Support
  - Footprint

Utypes properties (1)

- Consistency (to improve)
  - A utype relates to a well defined model
- Readibility
  - A clear string helps to understand the meaning → mapping is easier for the newcomer
  - Pointer to more documented products : Utype string +doc

    → see the uri mecanism proposed by N.Gray and vocabulary definitions by semantic group.

Utypes properties (2)

- Unicity within a document (cf. F. Ochsenbein)
  - We cannot avoid cases where a utype may be used several time
    - Ex: A complex Observation with several segments varying with time : **cha:TimeAxis.Coverage.Location** appears for each segment
  - No simple solution
  - Let the server decide which data he delivers: he knows the context
  - Should be defined inside TAP protocol

To be defined here

- Syntax:

  Suggestion : reuse the UCD mecanism:

cha:characterisationAxis.Coverage.location ; stc:Position2D.Value2D.C1

Parent model 's class(context) ; leaf model (object)

Interpret backwards the model chain??

observation;characterisation;stc

- Length of Utypes
  - Long for humans, short for computer
  - Adopt internal abbreviation for local utypes management (Saada)
- Utypes search
  - Via a vocabulary description (cf semantics)
- Versioning

  Distinguish various data model versions via a namespace