

# Utypes : steps forward

Data Model Working Group , M.Louys, IVOA Interop Pune, Oct 2011



# What are Utypes for ?

---

- ▶ A hook to a **data model element**
- ▶ Brings some concept + vocabulary
- ▶ Used in order to project some knowledge ( the data model element definition) on a value or simple variable.
- ▶ Not only a semantic label: it also brings in definitions from the data model.
  - ▶ `obscore:Char.SpatialAxis.Resolution.refval` is defined in arcsec as a single value
  - ▶ `photdm:PhotometryFilter.DateValidityFrom` is given in ISO TIME format
- ▶ It reflects the objects structure and relationships



# History : a few milestones

- ▶ **2003**: the name 'Utype' appears during Boston Interop
    - ▶ A unique tag to point to an element of a data model
    - ▶ Various WG interested : Theory domain model, DM, VOTable
    - ▶ Syntax borrowed from Xpath : a utype is a path into a data model
    - ▶ If an xml schema exists , the Utype can simply be the path down to a leaf
  - ▶ **2004 -2005**: used in VOTable as a reference to an element into a model
    - ▶ In tabular data , it adds knowledge on the way some piece of metadata can be understood and used in the context of a data model
    - ▶ → **Interpretation**
    - ▶ Distinguished from UCDs, which on contrary assign a label corresponding to some class of metadata to a column name.
    - ▶ → **Classification**
    - ▶ Drawbacks identified: strings too long, can be ambiguous
- ▶

# History (2)

- ▶ **2006** Baltimore Interop : some models reuse-elements from others
  - ▶ A utype refers to a name space like for XML documents
  - ▶ Utypes concatenates (cf next slide)
  - ▶ Data model design is important: (G.Lemson, others)  
Simplified data models work better: avoid templates , avoid substitution groups in XSD
- ▶ **2007** Utypes as uri : proposal for a syntax and use , N. Gray
- ▶ **2008** A. Micol : User's guide to map Characterisation utypes on archive metadata  
[http://www.ivoa.net/internal/IVOA/CharacLegacyPage/ivoa\\_char\\_2d\\_image\\_tutorial\\_1.0.pdf](http://www.ivoa.net/internal/IVOA/CharacLegacyPage/ivoa_char_2d_image_tutorial_1.0.pdf)
- ▶ **2009** How to publish and check utypes N.Gray *rdf based*  
<http://nxg.me.uk/note/2009/utype-proposals/>  
*IVOA Working draft* : [VD-Utypes-0.4-20091107.pdf](#)
- ▶ **2010** Utypes are widely used in protocols, TAP schema and serialisation
- ▶ **2011** Utypes planned to be used in PQL



# Starting from an XML Example

XML serialisation

CharacterisationDM

```
<characterisationAxis>
  <axisName>spatial</axisName>
  <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

Name space

# Utype syntax definition

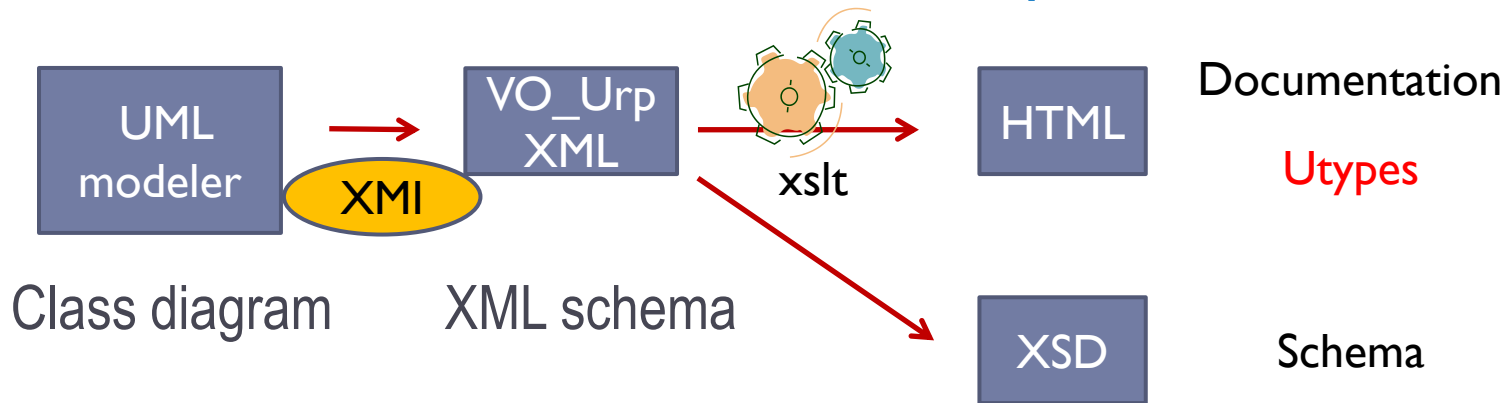
---

- ▶ A path to an element of a data model: from package name down to the most internal attribute.
- ▶ Utype strings defined ‘manually’ for first models , like Characterisation, SpectralDM, SLAP, etc.
- ▶ Published as a list of valid Utype strings (in or outside the standard document)
  - ▶ Spectrum/SSA
  - ▶ Characterisation
  - ▶ Can do better:
  - ▶ Create searchable lists referenced on IVOA repository like associated XML schema



# Utype automatic generation

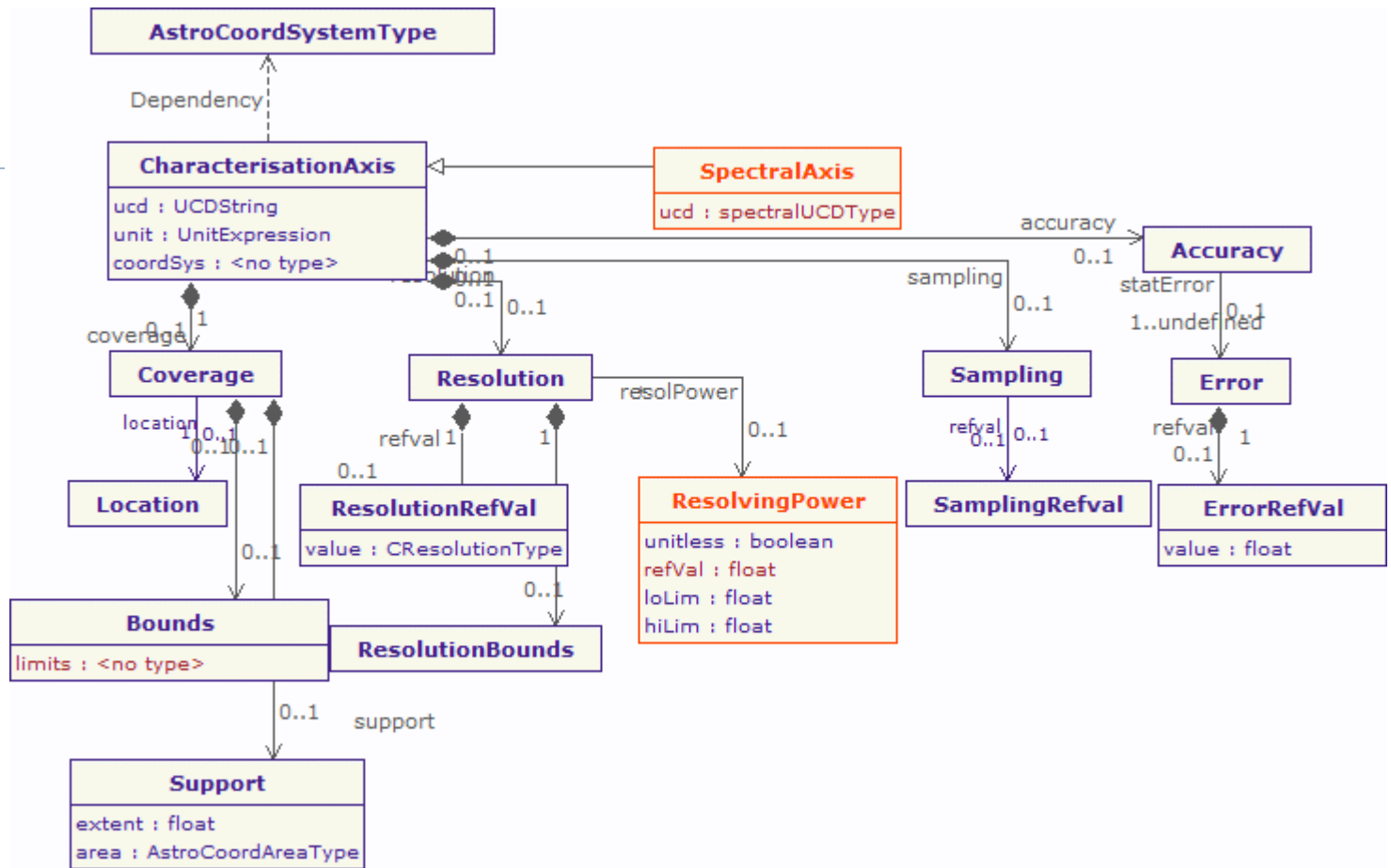
- ▶ Current models : use an automated process



- ▶ Apply **VO\_URP** the extraction pipeline ( G.Lemson et coll. )
- ▶ <http://volute.googlecode.com/svn/trunk/projects/theory/snapdm/specification/xslt/>
- ▶ Caveats:
  - ▶ XMI not standard between modelers. Needs tuning
- ▶ **Use some reasonable design rules for UML representation**



# UML rules



- ▶ Attribute name start with lower case
- ▶ Classes and attribute names may use CamelCase
- ▶ Prefer reference to inclusion
- ▶ Avoid nesting of too many packages → shorter utypes



# Reuse classes from a different model

---

- ▶ Include the reusable parts of existing DMs :
- ▶ ObsTAP re-uses classes from Spectrum DM: [Curation](#), [DataID](#), [Target](#), and from SSA : [Access class](#)
- ▶ These objects are directly included in the model and reused in a specific way defined in the context of the UML model
  - ▶ Follow down the encapsulated attributes within reused classes
  - ▶ Use only one name space
- ▶ Include reusable types
- ▶ Include classes from other models as types of attributes ( STC re-use)
  - ▶ Avoid long utypes with package prefix
  - ▶ Avoid concatenation of name spaces

[Characterisation.SpatialAxis.Coverage.Bounds.limits.Interval.LoLimit2Vec.C1](#)



# Utype usage

- ▶ **Goal:** To validate utype strings towards a DM
  - ▶ Compatibility is built-in when defining a TAP SCHEMA : mapping
  - ▶ Serialisations are not checked
- ▶ **How to validate utype used in VOTable PARAM or FIELD towards a DM**
- ▶ **Could use 3 steps:**
  - ▶ Syntactical analysis → a grammar ( see next slide)
  - ▶ Compare strings
  - ▶ Check Type/value compatibility
  - ▶ Check Type/value compatibility Units
  - ▶ **Is it difficult to include in a VOTable parser?**
- ▶ **Used in applications :**
  - ▶ Should be readable for developers → Camel Case preferred
  - ▶ Can be compared in lowercase for better efficiency (search)



# BNF grammar definition

G. Lemson et al.,

[http://volute.googlecode.com//svn/trunk/projects/theory/snapdm/specification/uml/DataModel\\_Profile.doc](http://volute.googlecode.com//svn/trunk/projects/theory/snapdm/specification/uml/DataModel_Profile.doc)

utype	:=	[model-utype   package-utype   class-utype   attribute-utype   collection-utype   reference-utype   container-utype ]
model-utype	:=	<model-name>
package-utype	:=	model-utype “:” package-hierarchy
package-hierarchy	:=	<package-name> [“/” <package-name>]*
class-utype	:=	package-utype “/” <class-name>
attribute-utype	:=	class-utype “.” attribute
attribute	:=	[primitive-attr   struct-attr]
primitive-attr	:=	<attribute-name>
struct-attr	:=	<attribute-name> “.” attribute
collection-utype	:=	class-utype “.” <collection-name>
reference-utype	:=	class-utype “.” <reference-name>
container-utype	:=	class-utype “.” “CONTAINER”
identifier-utype	:=	class-utype “.” “ID”



# Plans

---

- ▶ Revisit the existing Utype WD and converge on an updated version

<http://www.ivoa.net/cgi-bin/twiki/bin/view/IVOA/Utypes>

- ▶ Publish Utype lists as searchable documents
  - ▶ Xml enumerated list [UtypeChar2.0partiallist.xsd](#)

