Convergence between utypes and vodml-roles

F.Bonnarel (CDS and DAL chair)

Acknowledgments :
M.louys and L.Michel for discussions

VO services response typology

- VOTABLE Responses of services appear in three different use contexts :
  - 1) fully standardized responses with accurate definitions of fields
    - DAL discovery query responses (ObsTAP/ObsCORE, SIA, SSA, SimDAL..)
  - 2) catalog or tables (catalogs of sources , log of observation) where most of the content and fields is unstandardized
    - Some columns play a definite role (photometry, coordinates in space, access details…) described as in context 1
  - 3) Extended and hierarchised metadata or data. Client and servers have to exchange data model instances
    - "VODML mapping into VOTABLE" is the current proposed answer for VOTABLE
    - In DAL, extended metadata or data retrieval belongs to context 3

More on the 3 contexts distinction

- Context 3 « annotation » is a strong usage of datamodels
  - VO Software elements are chained and share the same datamodel instances.
  - « advised » clients or services.
- In context 1 and 2 , clients are less aware of full data model structure
  - They essentially consider each FIELD individually
  - Look for a given « ivoa role » of FIELDS to decide appropriate behavior

More on the 3 contexts distinction

Context 1 and 2 also distinguish strongly

- In context 1, we have accurate definition of the ivoa role of each VOTABLE FIELD/PARAM.
  - In practice definitions given in ivoa specifications.
  - In the case of DAL protocols lists of ivoa roles defined in the document.
  - VOTABLE utype attribute generally used for containing « ivoa roles ».
  - client software essentially « waiting for » well defined structuration of the upcoming data
- In context 2, FIELDS are tagged each time it is pertinent
  - Columns containing dataset fov description tagged with utype « obs:char.spatial.coverage.support.Area »
  - Columns contaning url and format of a dataset tagged with utype « Access.access_reference and « access.format »
  - Columns containing photometric values , filters , zero points are tagged with photometry model utypes
  - Unfortunately no official « stc » list
  - Client software looking for an « ivoa role » of FIELDS in a predefined list and adapt behavior if they find them

Focus on a question :

is VOTABLE utype attribute really usefull ?

- Could names or ucd be sufficient to specify « ivoa roles » ?
- Names :
  - In the case of ObsCore table
    - ObsCore is exposed in TAP services
    - The same ADQL query has to work on all TAP services
    - Names are defined by the specification because ADQL doesn't use utypes
  - In all other cases (other DAL protocols or context 2) good interoperabilty requires original « names » to stay unchanged

Focus on a question :

is VOTABLE utype attribute really usefull

- UCDs
  - Ucd are standard definition of nature of column content.
  - Sometime fuzzy (to allow comparisons across columns)
  - UCD combination may help to define role but not generally
    - Fuzzy meaning + fuzzy meaning not always gives « accurate »
  - Example :
    - The same table contains an observation target position and the location of the dataset in space
    - Both fields have ucd « pos.eq ».
    - They differ in role and also maybe in values
    - « pos.eq;src » could help for the target, but nothing exist for the « location »
    - ----> that's why utypes were invented in VOTABLE 1.2

A repository for utypes

- Utypes definitions are currently dispersed in various documents and places
- Need for unification and availaibility
- IVOA should maintain a list of those in a formatted document
  - semantics ?
  - DM ?
  - DAL ? ---> volunteers

What is the mapping between « ivoa roles » and data models

- Computer science defines « roles » and relationship with objet data models
- In « ORM » roles ares seen as predicates affecting entities in « facts » (sentences)
  - Entities can be groupped in classes
  - Roles separate in relationship/attributes +...
  - ….values
- See my Trieste presentation for more details :

http://wiki.ivoa.net/internal/IVOA/InteropOct2016DM/ivoa_roles.pdf

- Important property for us is that simple roles may be chained/combined in more complex roles
  - As we look for individual FIELDS « ivoa roles », it is important to distinguish them
    - The stat error of the Target position has a different « ivoa role » than a stat error in location position of the dataset

The mapping in practice (1)

- Current situation:
  - All datamodels used in DAL protocols so far have list of utypes
    - ObsTAP and SIAV2 ->ObsCore utypes,
    - SLAP -> SSLDM utypes,
    - SSA -> spectrum 1.0 utypes
    - SimDAL -> SimDM utypes
- Create the ivoa repository as an xml document containing
  - Utypes strings
  - One sentence of definition
  - Pointer to an appropriate specification document, section and page
  - Link to appropriate vodml-xml feature when available

The mapping in practice (2)

- From now onwards ivoa datamodels have a standard vodml-xml representation :
  - A well defined and interoperable universal representation
  - Each model is reprsented as an xml document
  - Each object, attribute, reference has a « vodml-role » stored in « vodml-id » element
  - If they have simple types, attributes/leaves may correspond to a FIELD in a votable
    - In that case a utype will be equivalent to a vodml-role
  - If attributes/leaves have complex types defined elsewhere in the datamodel or in another one
    - Needs to follow the path given by the vodml-type of the intermediary leaves
    - A complex « ivoa role » will be rendered by a sequence of vodml roles

Vodml xml : What does it look like ?

- Excerpt of Source Toy model vodml-xml document (thanks to Laurent Michel) :

```
<package>
<vodml-id>source</vodml-id><name>source</name><description>
      TODO : Missing description : please, update your UML model asap.
    </description>
    <objectType>
        <vodml-id>source.Source</vodml-id><name>Source</name><description>
        TODO : Missing description : please, update your UML model asap.
        </description>
        <attribute>
            <vodml-id>source.Source.name</vodml-id><name>name</name><description>
           TODO : Missing description : please, update your UML model asap.
            </description>
            <datatype><vodml-ref>ivoa:string</vodmlref>
           </datatype><multiplicity><minOccurs>1</minOccurs><maxOccurs>1</maxOccurs></multiplicity></attribute>
        <attribute>
            <vodml-id>source.Source.position</vodml-id><name>position</name><description>
            TODO : Missing description : please, update your UML model asap.
            </description>
            <datatype><vodml-ref>coords_tessel:domain.spatial.Position2D</vodml-ref>
            </datatype><multiplicity><minOccurs>1</minOccurs><maxOccurs>1</maxOccurs></multiplicity></attribute>
```

Vodml-roles

Simple type

Complex type

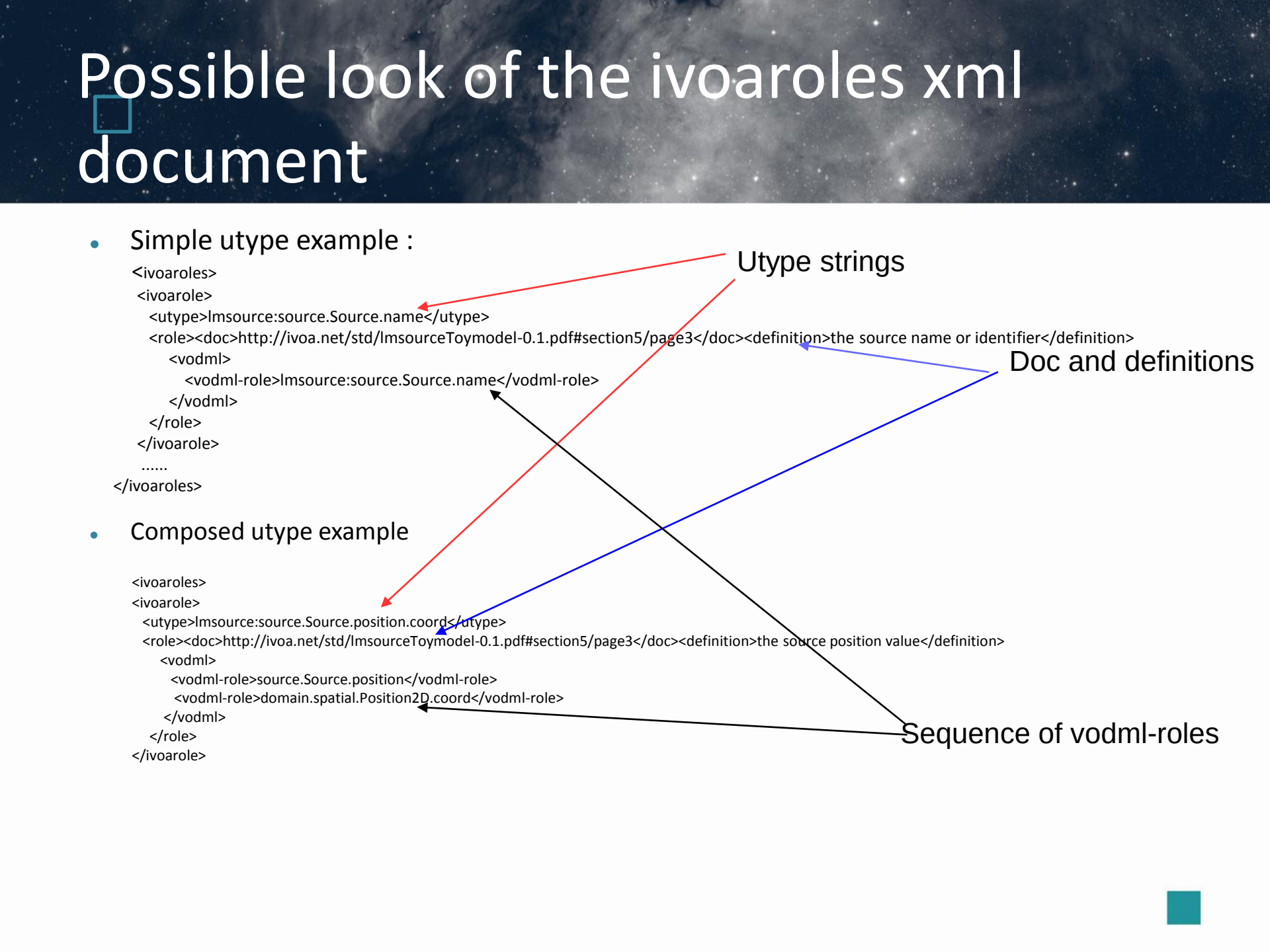
Possible look of the ivoaroles xml document

- Simple utype example :

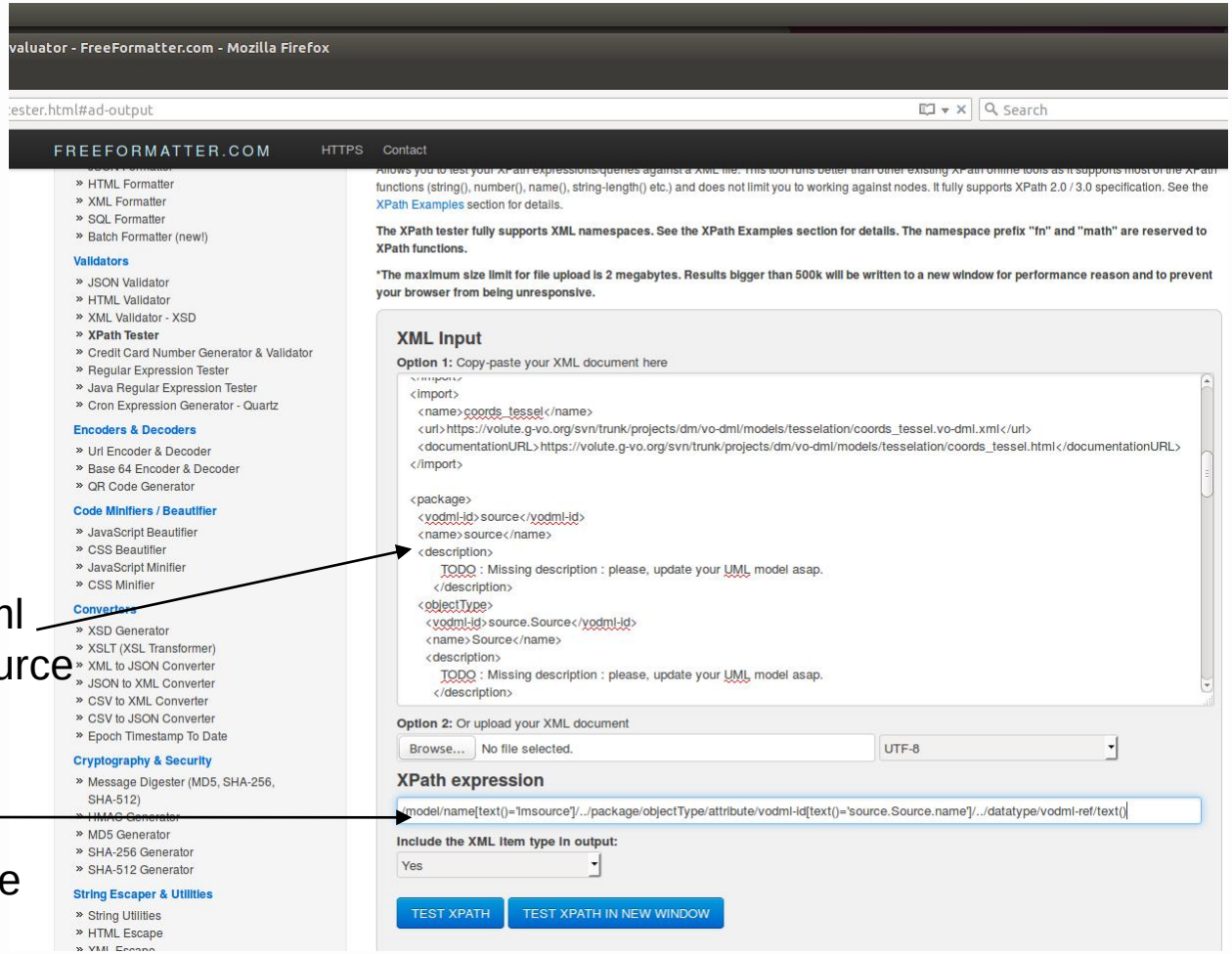
```
<ivoaroles>
 <ivoarole>
   <utype>lmsource:source.Source.name</utype>
   <role><doc>http://ivoa.net/std/lmsourceToymodel-0.1.pdf#section5/page3</doc><definition>the source name or identifier</definition>
      <vodml>
         <vodml-role>lmsource:source.Source.name</vodml-role>
      </vodml>
   </role>
 </ivoarole>
  ......
</ivoaroles>
```

- Composed utype example

```
<ivoaroles>
<ivoarole>
  <utype>lmsource:source.Source.position.coord</utype>
  <role><doc>http://ivoa.net/std/lmsourceToymodel-0.1.pdf#section5/page3</doc><definition>the source position value</definition>
     <vodml>
       <vodml-role>source.Source.position</vodml-role>
        <vodml-role>domain.spatial.Position2D.coord</vodml-role>
     </vodml>
  </role>
</ivoarole>
```



More with the vodml-xml document : xpathes

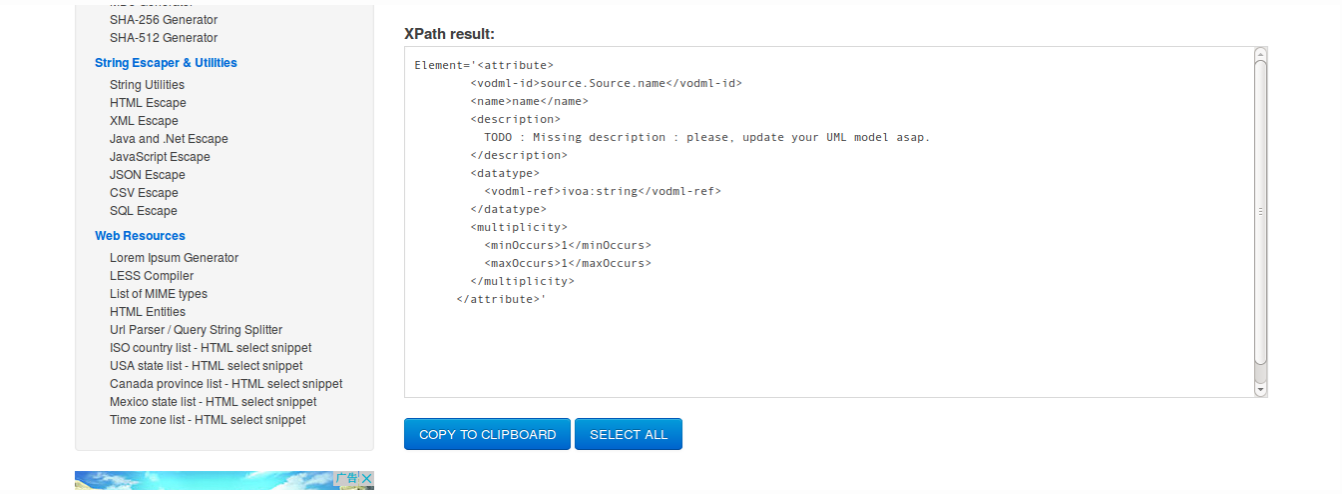


Vodml-xml
For lmsource

Xpath
For name
attribute

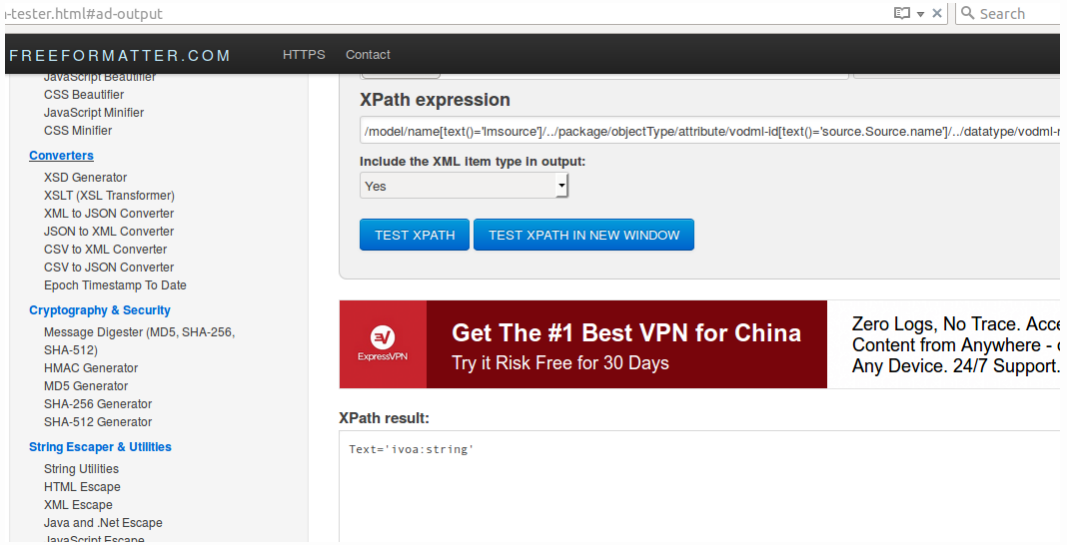
More with the vodml-xml document xpathes

- Result : name attribute



More with the vodml-xml document xpathes

Attribute datatype



Attribute description