

# Convergence between utypes and vodml-roles



---

F.Bonnarel (CDS and DAL chair)

Acknowledgments :

M.louys and L.Michel for discussions



# VOTABLE 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 containing 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 interoperability requires original « names » to stay unchanged



# Focus on a question :

is VOTABLE utype attribute really useful

- 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 availability
- 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 object data models
- In « ORM » roles are seen as predicates affecting entities in « facts » (sentences)
  - Entities can be grouped 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](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 represented 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</vodml-ref>
  </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>
</objectType>
</package>
```

Vodml-roles

Simple type

Complex type



# Possible look of the ivoaroles xml document

- Simple utype example :

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

Utype strings

Doc and definitions

- Composed utype example

```
<ivoaroles>
<ivoarole>
  <utype>Imsource:source.Source.position.coord</utype>
  <role><doc>http://ivoa.net/std/ImsourceToymodel-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>
```

Sequence of vodml-roles



# More with the vodml-xml document : xpathes

FreeFormatter.COM | HTTPS | Contact

- » HTML Formatter
- » XML Formatter
- » SQL Formatter
- » Batch Formatter (new!)

**Validators**

- » JSON Validator
- » HTML Validator
- » XML Validator - XSD
- » **XPath Tester**
- » Credit Card Number Generator & Validator
- » Regular Expression Tester
- » Java Regular Expression Tester
- » Cron Expression Generator - Quartz

**Encoders & Decoders**

- » Url Encoder & Decoder
- » Base 64 Encoder & Decoder
- » QR Code Generator

**Code Minifiers / Beautifier**

- » JavaScript Beautifier
- » CSS Beautifier
- » JavaScript Minifier
- » CSS Minifier

**Converters**

- » XSD Generator
- » XSLT (XSL Transformer)
- » XML to JSON Converter
- » JSON to XML Converter
- » CSV to XML Converter
- » CSV to JSON Converter
- » Epoch Timestamp To Date

**Cryptography & Security**

- » Message Digester (MD5, SHA-256, SHA-512)
- » HMAC Generator
- » MD5 Generator
- » SHA-256 Generator
- » SHA-512 Generator

**String Escaper & Utilities**

- » String Utilities
- » HTML Escape
- » XML Escape

Allows you to test your XPath expressions/queries against a XML file. This tool runs better than other existing XPath online tools as it supports most of the XPath functions (string(), number(), name(), string-length() etc.) and does not limit you to working against nodes. It fully supports XPath 2.0 / 3.0 specification. See the [XPath Examples](#) section for details.

The XPath tester fully supports XML namespaces. See the XPath Examples section for details. The namespace prefix "fn" and "math" are reserved to XPath functions.

\*The maximum size limit for file upload is 2 megabytes. Results bigger than 500k will be written to a new window for performance reason and to prevent your browser from being unresponsive.

**XML Input**

Option 1: Copy-paste your XML document here

```
<import>
<name>coords_tessel</name>
<url>https://volute.g-vo.org/svn/trunk/projects/dm/vo-dml/models/tesselation/coords_tessel.vo-dml.xml</url>
<documentationURL>https://volute.g-vo.org/svn/trunk/projects/dm/vo-dml/models/tesselation/coords_tessel.html</documentationURL>
</import>

<package>
<vodml-id> source</vodml-id>
<name> source</name>
<description>
  TODOQ : Missing description : please, update your UML model asap.
</description>
<objectType>
<vodml-id> source.Source</vodml-id>
<name> Source</name>
<description>
  TODOQ : Missing description : please, update your UML model asap.
</description>
</objectType>
</package>
```

Option 2: Or upload your XML document

Browse... No file selected. UTF-8

**XPath expression**

```
(model/name)[text()='Imsource']/../package/objectType/attribute/vodml-id[text()='source.Source.name']/../datatype/vodml-ref/text()
```

Include the XML Item type in output:

Yes

TEST XPATH TEST XPATH IN NEW WINDOW

Vodml-xml  
For Imsource

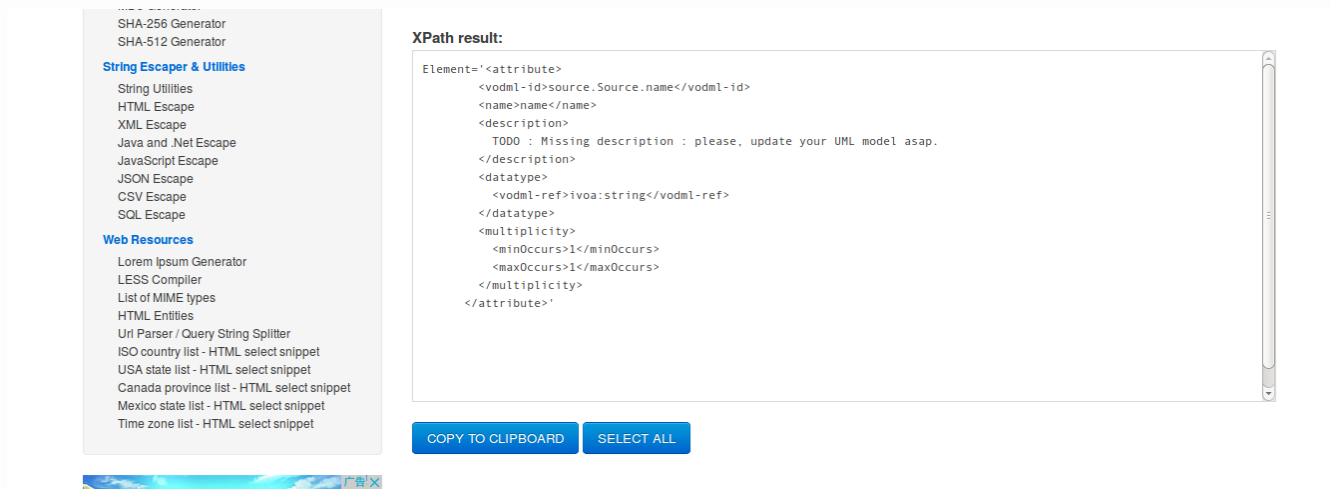
Xpath  
For name  
attribute





# More with the vodml-xml document xpathes

- Result : name attribute



The screenshot shows a web application interface. On the left is a sidebar menu with the following items:

- SHA-256 Generator
- SHA-512 Generator
- String Escaper & Utilities**
  - String Utilities
  - HTML Escape
  - XML Escape
  - Java and .Net Escape
  - JavaScript Escape
  - JSON Escape
  - CSV Escape
  - SQL Escape
- Web Resources**
  - Lorem Ipsum Generator
  - LESS Compiler
  - List of MIME types
  - HTML Entities
  - Uri Parser / Query String Splitter
  - ISO country list - HTML select snippet
  - USA state list - HTML select snippet
  - Canada province list - HTML select snippet
  - Mexico state list - HTML select snippet
  - Time zone list - HTML select snippet

On the right, an 'XPath result:' window displays the following XML snippet:

```
Element='<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</vodml-ref>
</datatype>
<multiplicity>
  <minOccurs>1</minOccurs>
  <maxOccurs>1</maxOccurs>
</multiplicity>
</attribute>'
```

Below the XPath result window are two buttons: 'COPY TO CLIPBOARD' and 'SELECT ALL'.





# More with the vodml-xml document xpathes

Attribute datatype

The screenshot shows the 'XPath expression' tool interface. The browser address bar contains 't-tester.html#fad-output'. The page header includes 'FREEFORMATTER.COM', 'HTTPS', and 'Contact'. The left sidebar lists various tools: JavaScript Beautifier, CSS Beautifier, JavaScript Minifier, CSS Minifier, Converters (XSD Generator, XSLT, XML to JSON, JSON to XML, CSV to XML, CSV to JSON, Epoch Timestamp To Date), Cryptography & Security (Message Digester, HMAC, MD5, SHA-256, SHA-512), String Escaper & Utilities (String Utilities, HTML, XML, Java and .Net, JavaScript, JSON, CSV, SQL), and Web Resources. The main content area shows the 'XPath expression' input field with the expression: `/model/name[text()='lmsource']/../package/objectType/attribute/vodml-id[text()='source.Source.name']/../datatype/vodml-`. Below the input is a dropdown menu for 'Include the XML item type in output' set to 'Yes'. Two buttons are present: 'TEST XPATH' and 'TEST XPATH IN NEW WINDOW'. Below the main content is a red banner for 'ExpressVPN' with the text 'Get The #1 Best VPN for China Try it Risk Free for 30 Days' and 'Zero Logs, No Trace. Account Content from Anywhere - on Any Device. 24/7 Support.' The 'XPath result:' section shows the output: `Text='ivoa:string'`.

Attribute description

The screenshot shows the 'XPath result:' section of the tool. The output is: `Element='<description>  
 TODO : Missing description : please, update your UML model asap.  
</description>'`. Below the output are two buttons: 'COPY TO CLIPBOARD' and 'SELECT ALL'.

