VIRTUAL ASTRONOMICAL OBSERVATORY

UTYPEs Working Draft

Omar Laurino
SAO/VAO
For the UTYPEs Tiger Team

Tiger Team mandate

- Collect Use Cases
- Study current usage of Utypes
  - Are they used in a meaningful way?
  - Assess impact of changes to existing standards
- Develop a standard

Tiger Team mandate

- Collect Use Cases **DONE (Urbana + SaoPaulo)**
- Study current usage of Utypes
  - Are they used in a meaningful way?
  - Assess impact of changes to existing standards
- Develop a standard

Tiger Team mandate

- Collect Use Cases
- Study current usage of Utypes **DONE (+updates)**
  - Are they used in a meaningful way?
  - Assess impact of changes to existing standards
- Develop a standard

Tiger Team mandate

- Collect Use Cases
- Study current usage of Utypes
  - Are they used in a meaningful way?
  - Assess impact of changes to existing standards
- Develop a standard **UTYPEs WD v1.0, Heidelberg**

Forerunners

- Louys & Bonnarel, May 2004 Note:
  - Data Model serialisation in VOTable
  - *a more simple "quick and dirty" proposal* (than pure XML)
  - *If we take into account that metadata description and structuration are part of the goals for a datamodel, we must consider the fact that such kind of description is some special case of a structured catalogue, where the columns are metadata, the lines are the instances of objects..*

Forerunners

- Derriere, May 2011 Talk in Naples (+Note):
  - *How to represent Data Model metadata in VOTable? Data Model has: Structure (hierarchy), utypes*
  - *<GROUP>s are your friends*
  - *Map!*
  - *One table row = one object. Model attribute represented as: PARAM if same value for all rows; FIELD if value in table column. ADD hierarchy of GROUP, PARAM, PARAMref, FIELDref with model utypes.*
  - *One VOTable can represent several DMs: different utypes in FIELD and FIELDref have different GROUPs with different utypes*

What the Tiger Team added

- Recursion
  - if an **Object** *is represented by a* **GROUP** => an **Object in an Object** *is represented by a* **GROUP in a GROUP**
- Standardization:
  - Data Modeling Language (meta-model)
  - Mapping Language from meta-model to meta-model: UTYPEs!
  - Rigorous Mapping Patterns specification
- Backwards compatibility w/ current/custom usages
- UTYPEs portability:
  - *If I can* **be an error bar** *there, I'll* **be it** *everywhere*
  - *If I can* **spot an error bar** *there, I'll* **spot it** *everywhere*

Liza "Data Model" Minnelli

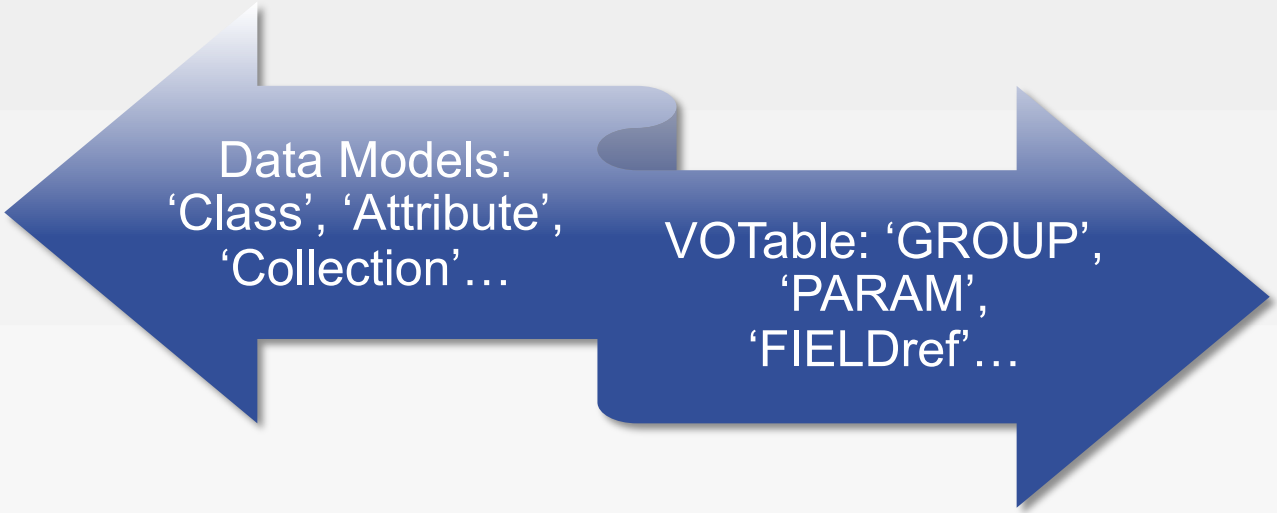
Utypes, Utypes

The Need For a Mapping Language

Everybody agrees: UTYPEs are pointers.
But pointers to what?

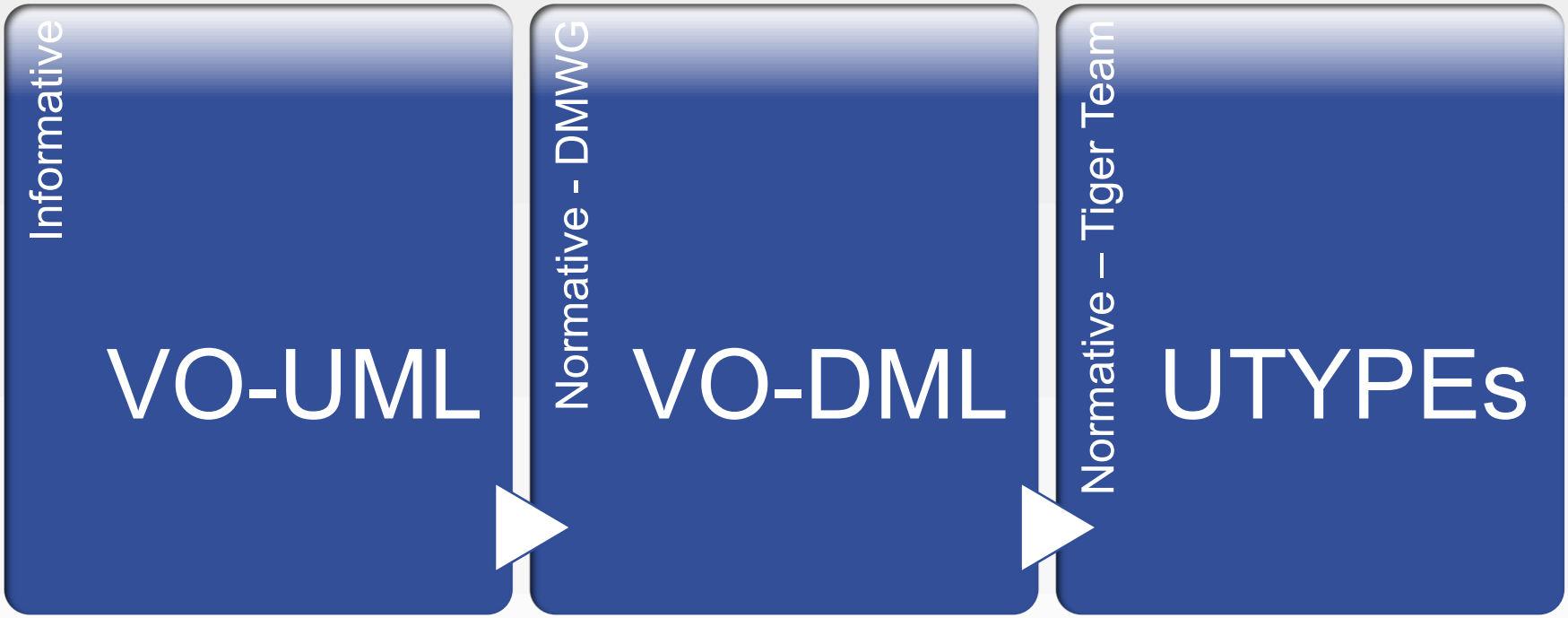
The Need For a Mapping Language

Everybody agrees: UTYPEs are pointers.
But pointers to what?



UTYPEs map concepts in VOTable, e.g. GROUP, PARAM to concepts in a meta-model, e.g. Class, Attribute.

The Big Picture



UTYPEs format

Everybody Agrees

- utype ::= prefix ':' localname

Alternative 1

- Prefixes are fixed for all DMs

Alternative 2

- Prefixes are fixed only for IVOA RECs
- To be guaranteed unique, Extensions' prefixes have a lifecycle limited to the single document

~~Alternative 3~~

- ~~All prefixes have a lifecycle limited to the single document~~

UTYPEs portability (where's the mug?)

- Problem: **tag objects with a labelmaker, e.g. a mug**

- Solution 1:

omar:Kitchen.Shelf.Mug

jesus:LivingRoom.CoffeeTable.Mug

…

- Solution 2:

Type = pottery:Mug

Role = omar:Shelf.mug (in omar:Kitchen)

Role = jesus:CoffeeTable.mug (in jesus:LivingRoom)

…

UTYPEs portability (where's the mug?)

- Problem: **Ask a robot to find a mug in Pat's house**

- Solution 1:

pat:DiningRoom.Table.Mug

- Solution 2:

Type = pottery:Mug

Role = pat:Table.mug (in pat:DiningRoom)

UTYPEs portability (where's the mug?)

- Problem: **Ask a robot to find a mug in Pat's house**

- Solution 1:

pat:DiningRoom.Table.Mug

It works w/ a formal Grammar, a Vocabulary, a Parser.

- Solution 2:

Type = pottery:Mug

Role = pat:Table.mug (in pat:DiningRoom)

It works.

In a perfect world

UTYPE should be an element with structure,
not an attribute (at least *role* and *type*), but…
We don't want to change VOTable.

In a perfect world

UTYPE should be an element with structure,
not an attribute (at least *role* and *type*), but…
We don't want to change VOTable.

We might workaround by concatenating
role and type UTYPEs, but…
We don't want to parse UTYPEs.
(what about UCDs ';'?)

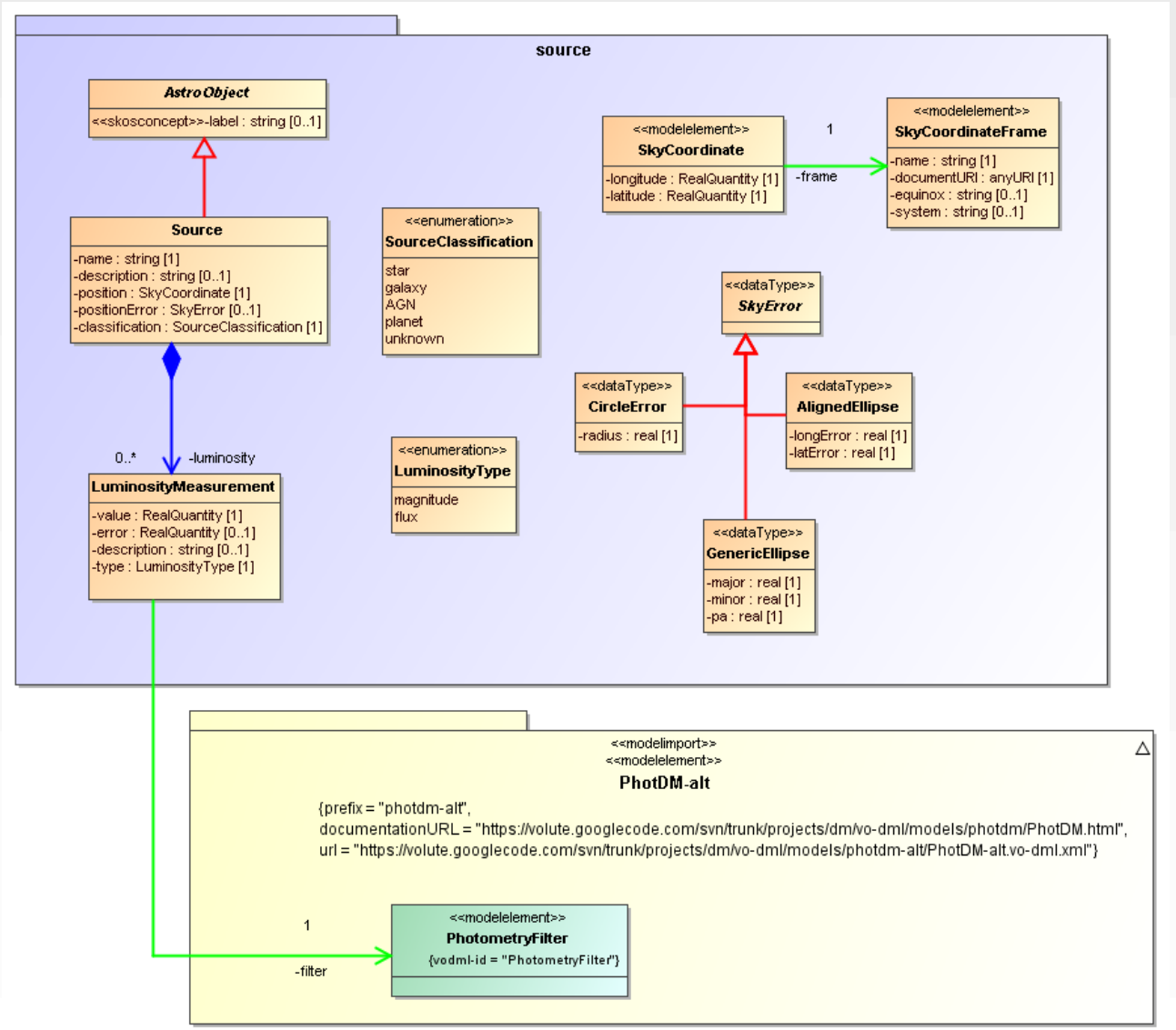
In a perfect world

UTYPE should be an element with structure,
not an attribute (at least *role* and *type*), but…
We don't want to change VOTable.

We might workaround by concatenating
role and type UTYPEs, but…
We don't want to parse UTYPEs.
(what about UCDs ';'?)

Falled back on a less compact, but *almost*
equivalent solution (with some unavoidable loss of information)

Examples


source
AstroObject
<<skosconcept>>-label : string [0..1]
Source
-name : string [1]
-description : string [0..1]
-position : SkyCoordinate [1]
-positionError : SkyError [0..1]
-classification : SourceClassification [1]
<<enumeration>>
SourceClassification
star
galaxy
AGN
planet
unknown
<<modelelement>>
SkyCoordinate
-longitude : RealQuantity [1]
-latitude : RealQuantity [1]
1
-frame
<<modelelement>>
SkyCoordinateFrame
-name : string [1]
-documentURI : anyURI [1]
-equinox : string [0..1]
-system : string [0..1]
<<dataType>>
SkyError
<<dataType>>
CircleError
-radius : real [1]
<<dataType>>
AlignedEllipse
-longError : real [1]
-latError : real [1]
<<dataType>>
GenericEllipse
-major : real [1]
-minor : real [1]
-pa : real [1]
0..*
-luminosity
LuminosityMeasurement
-value : RealQuantity [1]
-error : RealQuantity [0..1]
-description : string [0..1]
-type : LuminosityType [1]
<<enumeration>>
LuminosityType
magnitude
flux
<<modelimport>>
<<modelelement>>
PhotDM-alt
{prefix = "photdm-alt",
documentationURL = "https://volute.googlecode.com/svn/trunk/projects/dm/vo-dml/models/photdm/PhotDM.html",
url = "https://volute.googlecode.com/svn/trunk/projects/dm/vo-dml/models/photdm-alt/PhotDM-alt.vo-dml.xml"}
1
-filter
<<modelelement>>
PhotometryFilter
{vodml-id = "PhotometryFilter"}

Examples

Direct Serialization

- <GROUP utype="**vo-dml:Instance.root**">
  <PARAM utype="**vo-dml:Instance.type**" value="**src:source.Source**" />
  <**PARAM** utype="**src:source.Source.name**" value="08120809-0206132"/>
  <GROUP utype="**src:source.Source.position**">
  <PARAM utype="**vo-dml:Instance.type**" value="**src:source.SkyCoordinate**">
  <**PARAM** utype="**src:source.SkyCoordinate.longitude**" value="123.0337"/>
  <**PARAM** utype="**src:source.SkyCoordinate.latitude**" value="-2.103"/>
  <GROUP ref="_icrs" utype="**src:source.SkyCoordinate.frame**" />
  </GROUP>
  </GROUP>

Examples

Indirect Serialization

```
• <GROUP utype="vo-dml:Instance.root">
    <PARAM utype="vo-dml:Instance.type" value="src:source.Source" />
    <FIELDref ref="_designation" utype="src:source.Source.name" />
    <GROUP utype="src:source.Source.position">
      <PARAM utype="vo-dml:Instance.type" value="src:source.SkyCoordinate">
      <FIELDref ref="_ra" utype="src:source.SkyCoordinate.longitude" />
      <FIELDref ref="_dec" utype="src:source.SkyCoordinate.latitude" />
      <GROUP ref="_icrs" utype="src:source.SkyCoordinate.frame" />
    </GROUP>
  </GROUP>
```

Examples

Extensions

```
• <GROUP utype="vo-dml:Instance.root" >
   <PARAM utype="vo-dml:Instance.type" value="xsrc:source.Source"/>
   <PARAM utype="vo-dml:Instance.type" value="src:source.Source"/>
   <FIELDref ref="_designation" utype="src:source.Source.name"/>
   <FIELDref ref="_z" utype="xsrc:source.Source.redshift"/>
   <PARAM name="type" utype="src:source.Source.classification" value="galaxy">
   <GROUP utype="src:source.Source.position">
    <PARAM utype="vo-dml:Instance.type" value="src:source.SkyCoordinate"/>
    <FIELDref ref="_ra" utype="src:source.SkyCoordinate.longitude"/>
    <FIELDref ref="_dec" utype="src:source.SkyCoordinate.latitude"/>
    <GROUP ref="_icrs" utype="src:source.SkyCoordinate.frame"/>
   </GROUP>
  </GROUP>
```

Examples

If you are interested in more examples
we can look directly at the document

Plenty of Room (backwards compatibility)

Mappings not specified allow custom/legacy usage

- FIELDs.
- Standalone PARAMs.
- GROUPs with @utype of undeclared prefix.
- TABLE, RESOURCE

Transition

- No changes required in current stds, protocols.
- If Data Providers want to upgrade, they can **add** metadata, not change the current metadata.
- Services prototyped against WDs can simply **add** metadata.

New, complex Data Models can benefit

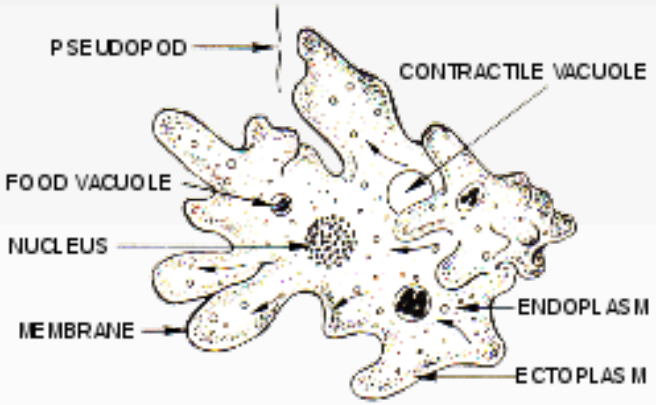
- Data nCubes and their projections/combinations: TimeSeries, SED, Spectral, Photometry

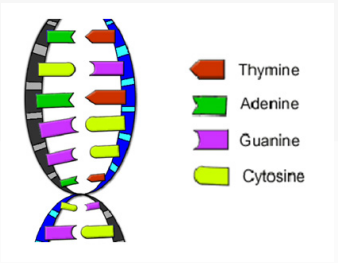
Growing complexity

Naïve client

- it does not parse the VO-DML description file
- it assumes the a priori knowledge of one or more Data Models
- it discovers information by looking for a set of predefined UTYPEs in the VOTable

Simple DAL Clients
Simple User scripts



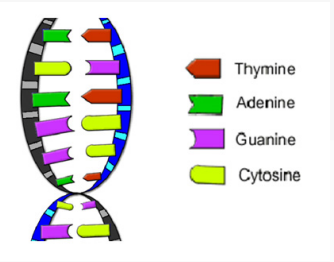


Growing complexity

Advanced Client

- it does not parse the VO-DML description file
- it is interested in the structure of the serialized instances
- can follow the mapping patterns defined in this specification, for example collections, references, and inheritance

Advanced DAL Clients
Science Applications
Smart Metadata Browsers

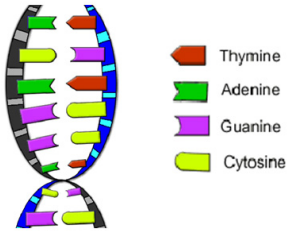


Growing complexity

Guru Client

- it parses the VO-DML descriptions
- it does not assume any a priori knowledge of any Data Models.
- Introspection/Reflection

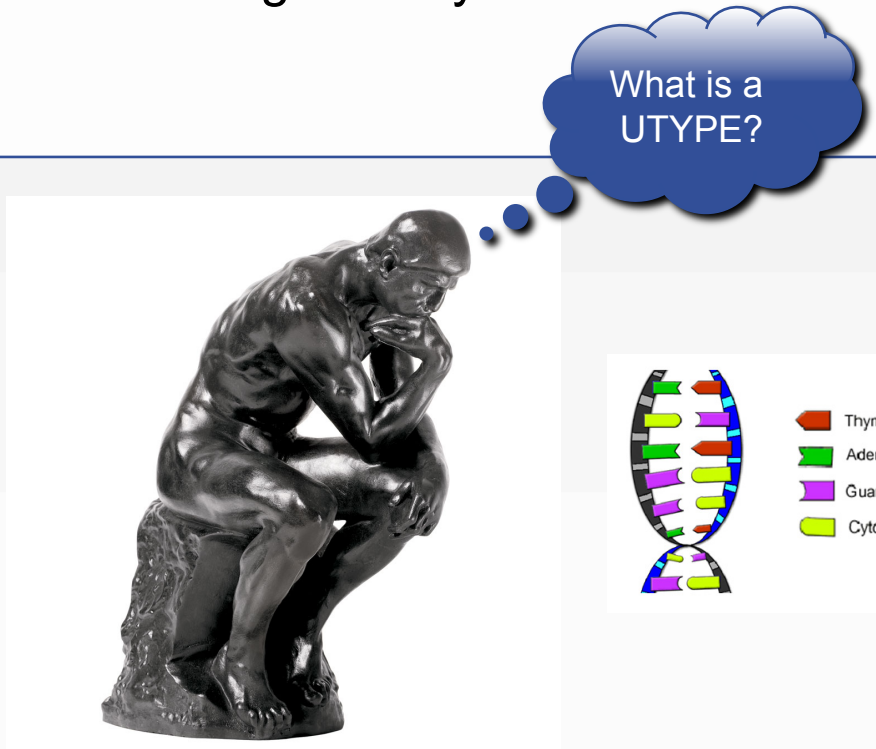
Universal Validators
Publishing Helpers
VO-Importer
Universal I/O Framework



Growing complexity

Guru Client

- it parses the VO-DML descriptions
- it does not assume any a priori knowledge of any Data Models.
- Introspection/Reflection

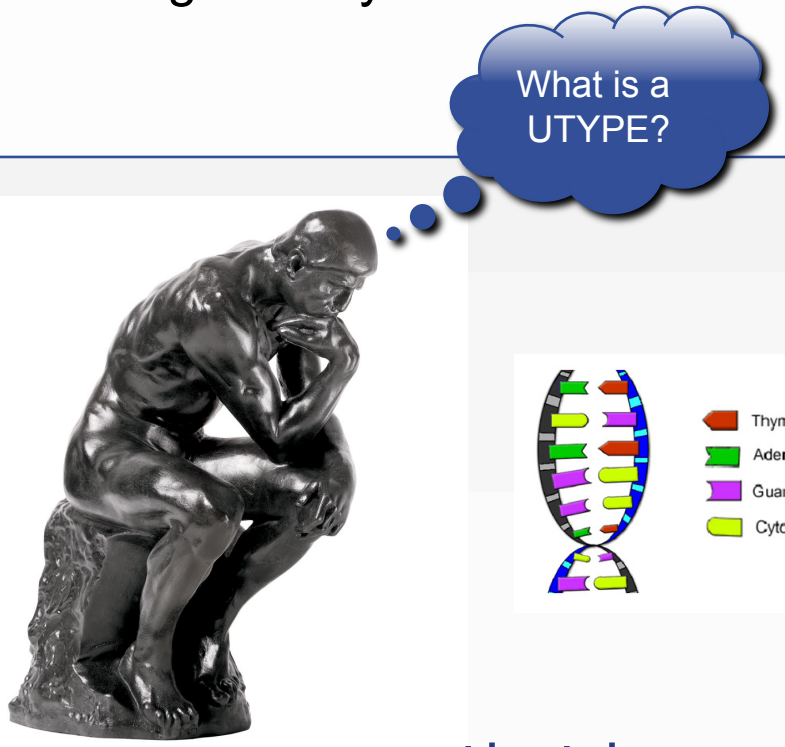


Universal Validators
Publishing Helpers
VO-Importer
Universal I/O Framework

Growing complexity

Guru Client

- it parses the VO-DML descriptions
- it does not assume any a priori knowledge of any Data Models.
- Introspection/Reflection



Universal Validators
Publishing Helpers
VO-Importer
Universal I/O Framework

Fun fact: amoebae have ~250x more genome that humans

Open Issues

- UTYPEs format: fixed or floating prefix?
- UTYPEs format: localname as URI fragment?

- GROUPref-s and PARAMref-s for VO-DML mapping?

- Object Relational Mapping: much work done, but…

- Serialization to formats other than VOTable

Need to get feedback from WGs (e.g. App) and community

Frequently Asked Questions

Check out the FAQ appendix in the WD
Answers to your doubts might be already there

Lessons Learned

- Tigers **are** an endangered species
- In the Facebook era, **nobody** likes Tigers
- Tigers can be aggressive
- So, next time…

Lessons Learned

- Tigers **are** an endangered species
- In the Facebook era, **nobody** likes Tigers
- Tigers can be aggressive
- So, next time…

Let's form a Kittens Team

Thank you!