

# Models and VOTables

~ Part I ~

Working with MIVOT

L. Michel

# ChatGPT encourages us: good guy!

Why do we need to map our data on standard data models in the virtual observatory 

Tl;dr:

Mapping data to standard data models in the virtual observatory helps ensure that data from different sources can be accessed, shared, and compared easily and efficiently. It also allows users to search for, manipulate, and analyze data more quickly and accurately.

**ChatGPT 12/2022**

# What do we expect from using VO Data Models

## What do models make easier

- Get a complete representation of data context
  - Provenance
  - Coordinate systems
  - Observation location
- Get an accurate representation of the errors
- Support cross-columns parameters
  - Columns grouping
  - Complex errors (pos + pm + parallax)
- Gather data from multi-table VOTable (sources + detection)
- Exchange model instance out of the VOTable context (e.g. SAMP)

# Moving toward Operational Implementation

- **Validator**
  - Validate annotated VOTable against the mapped models.
- **Documentation**
  - Help users to figure out what are the steps of the process
- **Tooling for the server side**
  - Showing up a generic implementation for a TAP service
- **Tooling for the client side**
  - Make a PR enabling AstroPyVO to work with annotated data.
- **Register MIVOT enable services**

These tools will help for both **implementing** services and **exercising model proposals on real data.**

# Let's have a look at a GAIA example

- **In the following we are using**
  - A regular TAP service
  - RECommended models (Meas, Coord)
  - MIVOT
- **On client side**
  - Code implemented for the MIVOT reference implementation
  - Astropy and PyVO
- **On server side**
  - Pseudo-code as we plan to develop for AstroPyVO
  - VOLLT extension Prototype (XTapDB)

# Resources

Repository: <https://github.com/ivoa/modelinstanceinvot-code/>  
Branch : `feature/instance_generator`

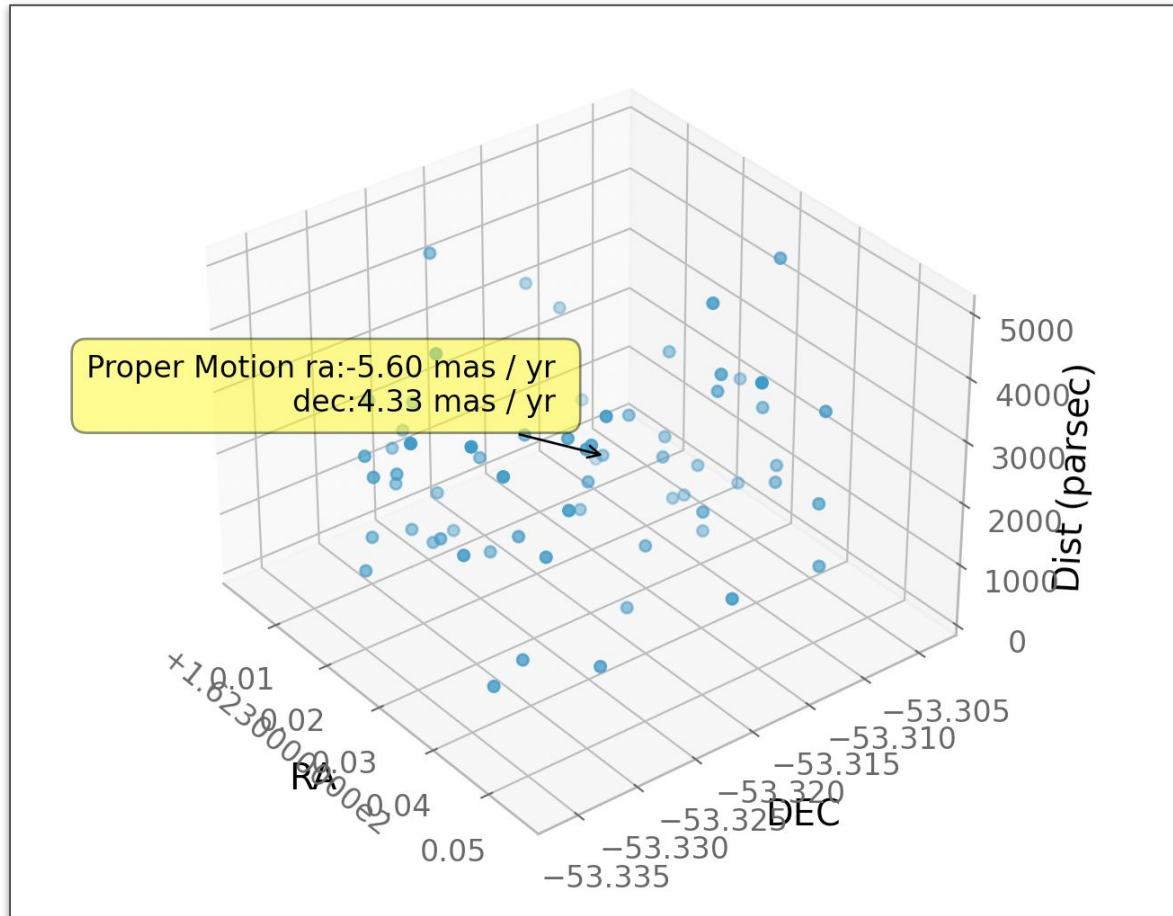
VOTABLE: [/mivot\\_code/examples/data/gaia\\_luhman16\\_astropy.xml](/mivot_code/examples/data/gaia_luhman16_astropy.xml)

CODE: </examples/example.4.astropy.py>

# Let's go around Luhman16

```
SELECT TOP 100 designation ,
               ra , ra_error , dec, dec_error
               parallax , parallax_error ,
               pmra , pmra_error, pmdec , pmdec_error ,
               radial_velocity , radial_velocity_error
FROM   gaiadr2.gaia_source ,
WHERE  ( CONTAINS (POINT('ICRS', ra, "dec"),
                  CIRCLE('ICRS', 162.328814, -53.319466
                        , 0.016)) = 1 )
```

# I want my code to do a 3D plot of the queried objects



**3D positions** are given by the combination of **RA/DEC** and the **parallax** (Measure model)

The **proper motion** is shown up as mouse-over **tooltips**

The **radial velocity** is **ignored** in this example



# The MIVOT Regular Structure

```
xsi:schemaLocation="http://www.ivoa.net/xml/VOTable/v1.3 http://www.ivoa.net/xml/VOTable/v1.3">
<RESOURCE type="results">
  <RESOURCE type="meta">
    <VODML xmlns:dm-mapping="http://www.ivoa.net/xml/merged-syntax">
      <REPORT status="OK">hand-made mapping</REPORT>

      <MODEL name="meas" url="https://www.ivoa.net/xml/Meas/20200908/Meas-v1.0.vo-dml.xml" />
      <MODEL name="coords" url="https://www.ivoa.net/xml/STC/20200908/Coords-v1.0.vo-dml.xml" />
      <MODEL name="ivoa" url="https://www.ivoa.net/xml/VODML/IVOA-v1.vo-dml.xml" />

      <GLOBALS>[]
      <TEMPLATES tabref="Results">[]
    </VODML>
  </RESOURCE>
  <INFO name="QUERY_STATUS" value="OK" />
  <INFO name="QUERY"
    value="SELECT TOP 100 gaiadr2.gaia_source.designation , gaiadr2.gaia_source.ra , gaiadr2.gaia_sourc
FROM gaiadr2.gaia_source
WHERE ( CONTAINS(POINT(&#039;ICRS&#039;;, ra, &quot;dec&quot;), CIRCLE(&#039;ICRS&#039;;, 162.328814, -53.
FROM gaiadr2.gaia_source
WHERE ( CONTAINS(POINT('ICRS', ra, "dec"), CIRCLE('ICRS', 162.328814, -53.319466, 0.016666666666666666))
  <INFO name="CAPTION"
    value="How to cite and acknowledge Gaia: https://gea.esac.esa.int/archive/documentation/credits.htm
  <INFO name="PAGE" value="" />
  <INFO name="PAGE_SIZE" value="" />
  <INFO name="USER_AGENT" value="" />
```

# Expand GLOBALS: the Space Frame

## Class of the Coordinates models

```
<GLOBALS>
  <INSTANCE dmid="SpaceFrame_ICRS" dmtpe="coords:SpaceSys">
    <INSTANCE dmrole="coords:PhysicalCoorusys.frame" dmtpe="coords:SpaceFrame">
      <ATTRIBUTE dmrole="coords:SpaceFrame.spaceRefFrame" dmtpe="ivoa:string" value="ICRS" />
      <INSTANCE dmrole="coords:SpaceFrame.refPosition" dmtpe="coords:CustomRefLocation">
        <ATTRIBUTE dmrole="coords:CustomRefLocation.epoch" dmtpe="coords:Epoch" value="2015.0"/>
      </INSTANCE>
    </INSTANCE>
  </INSTANCE>
</GLOBALS>
```

# Mapping of the Results table

Each row of the table `Results` will be mapped as an instance of class `Position` of the `Measure` model, plus other objects that are hidden on this view graph

```
<TEMPLATES tableref="Results">
```

```
<INSTANCE dmrole="" dmtpe="meas:Position">  
  <ATTRIBUTE dmrole="meas:Measure.ucd" dmtpe="ivoa:string" value="pos" />  
  <INSTANCE dmrole="meas:Measure.coord" dmtpe="coords:LonLatPoint">  
    <ATTRIBUTE dmtpe="ivoa:RealQuantity" dmrole="coords:LonLatPoint.lon" ref="ra" unit="deg"/>  
    <ATTRIBUTE dmtpe="ivoa:RealQuantity" dmrole="coords:LonLatPoint.lat" ref="dec" unit="deg"/>  
    <ATTRIBUTE dmtpe="ivoa:RealQuantity" dmrole="coords:LonLatPoint.dist" ref="parallax" unit="mas"/>  
    <REFERENCE dmrole="coords:Coordinate.coordSys" dmref="SpaceFrame_ICRS" />  
  </INSTANCE>  
  
  <INSTANCE dmrole="meas:Measure.error" dmtpe="meas:Asymmetrical3D">  
    <COLLECTION dmrole="meas:Asymmetrical3D.plus">  
      <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="ra_error" unit="mas"/>  
      <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="dec_error" unit="mas"/>  
      <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="parallax_error" unit="mas"/>  
    </COLLECTION>  
    <COLLECTION dmrole="meas:Asymmetrical3D.minus">  
      <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="ra_error" unit="mas"/>  
      <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="dec_error" unit="mas"/>  
      <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="parallax_error" unit="mas"/>  
    </COLLECTION>  
  </INSTANCE>  
</INSTANCE>
```

# Position + error made of RA/DEC + parallax

position

```
<INSTANCE dmrole="" dmtpe="meas:Position">
<ATTRIBUTE dmrole="meas:Measure.ucd" dmtpe="ivoa:string" value="pos" />
<INSTANCE dmrole="meas:Measure.coord" dmtpe="coords:LonLatPoint">
  <ATTRIBUTE dmtpe="ivoa:RealQuantity" dmrole="coords:LonLatPoint.lon" ref="ra" unit="deg"/>
  <ATTRIBUTE dmtpe="ivoa:RealQuantity" dmrole="coords:LonLatPoint.lat" ref="dec" unit="deg"/>
  <ATTRIBUTE dmtpe="ivoa:RealQuantity" dmrole="coords:LonLatPoint.dist" ref="parallax" unit="mas"/>
  <REFERENCE dmrole="coords:Coordinate.coordSys" dmref="SpaceFrame_ICRS" />
</INSTANCE>
```

Error on position

```
<INSTANCE dmrole="meas:Measure.error" dmtpe="meas:Asymmetrical3D">
  <COLLECTION dmrole="meas:Asymmetrical3D.plus">
    <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="ra_error" unit="mas"/>
    <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="dec_error" unit="mas"/>
    <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="parallax_error" unit="mas"/>
  </COLLECTION>
  <COLLECTION dmrole="meas:Asymmetrical3D.minus">
    <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="ra_error" unit="mas"/>
    <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="dec_error" unit="mas"/>
    <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="parallax_error" unit="mas"/>
  </COLLECTION>
</INSTANCE>
</INSTANCE>
```

# Proposal for PyVO extension (proto working)

```
# Read the VOTable as usual
votable = parse(os.path.join(data_path, "gaia_luhman16_astropy.xml"))
# Read the VOTable through the model view
mviewer = ModelViewer(votable)

while True:
    # get a data row (numpy)
    row = mviewer.get_next_row()
    if row is None:
        break
    # get the astropy SkyCoord object for that row
    position = mviewer.get_astropy_sky_coord()
    # stores the parameters to be plotted
    # Ignore not set-distances for the plot readability
    if not np.isnan(position.distance.value):
        ras.append(position.ra.value)
        decs.append(position.dec.value)
        dist.append(position.distance.value)
        tooltips.append(
            f"Proper Motion ra:{position.pm_ra_cosdec:.2f} \ndec:{position.pm_dec:.2f} "
        )

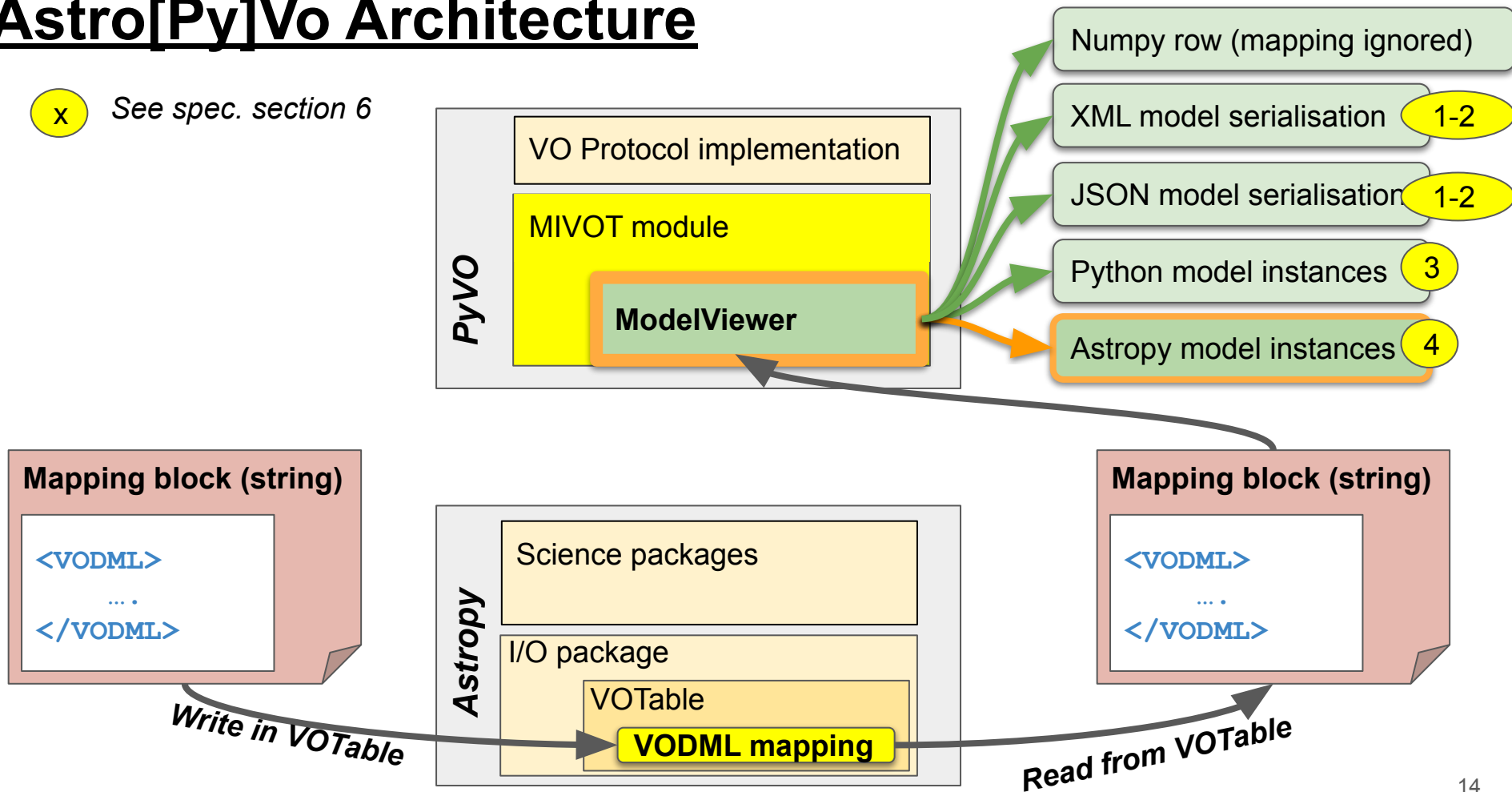
# do matplotlib stuff
```

The **model viewer** of the MIVOT PyVO extension (prototype) is able to **generate instances of `astropy.SkyCoord`** from 5 parameters

- Position
- Parallax
- Proper motion

# Astro[Py]Vo Architecture

x See spec. section 6



# Annoter on the server side (pseudo code)

```
service = pyvo.dal.TAPService(service_url)
result = service.run_sync(
    """SELECT TOP 100 designation ,
        ra , ra_error , dec, dec_error
        parallax , parallax_error ,
        pmra , pmra_error, pmdec , pmdec_error ,
        radial_velocity , radial_velocity_error
FROM    gaiadr2.gaia_source ,
WHERE ( CONTAINS(POINT('ICRS', ra, "dec"),
                CIRCLE('ICRS', 162.328814, -53.319466, 0.016)) = 1 )
"""
)
votable = result.votable
mivot_block = votable.addMivotBlock()
pos_id = mivot_block.addLonLat()
pos_id.mapColumn({"lon": "ra", "lat": "dec", "dist": "parallax"})
error_id = pos_id.addError("Asymmetrical3D")
error_id.mapColumn(
    {
        "plus": ["ra_error", "dec_error", "parallax_error"],
        "minus": ["ra_error", "dec_error", "parallax_error"],
    }
)
votable.sendBackToTheClient()
```

Get the VOTable by whatever mean

Build the annotation element per element

# Build an empty mapping block connected with the Results table

```
mivot_block = votable.addMivotBlock("Results", "mapped for Interop Bologna")
```

```
<RESOURCE type="results">  
  <RESOURCE type="meta">  
    <VODML xmlns:dm-mapping="http://www.ivoa.net/xml/merged-syntax">  
      <REPORT status="OK">mapped for Interop Bologna</REPORT>  
      <GLOBALS>  
      </GLOBALS>  
  
      <TEMPLATES tableref="Results">  
      </TEMPLATES>  
  
    </VODML>  
  </RESOURCE>
```



# Add a position instance + space frame

```
pos_id = mivot_block.addLonLat("ICRS", epoch=2015.0)
```

```
<VODML xmlns:dm-mapping="http://www.ivoa.net/xml/merged-syntax">
  <REPORT status="OK">mapped for Interop Bologna</REPORT>

  <MODEL name="meas" url="https://www.ivoa.net/xml/Meas/20200908/Meas-v1.0.vo-dml.xml" />
  <MODEL name="coords" url="https://www.ivoa.net/xml/STC/20200908/Coords-v1.0.vo-dml.xml" />
  <MODEL name="ivoa" url="https://www.ivoa.net/xml/VODML/IVOA-v1.vo-dml.xml" />

  <GLOBALS>
    <INSTANCE dmid="SpaceFrame_ICRS" dmtype="coords:SpaceSys">
      <INSTANCE dmrole="coords:PhysicalCoordSys.frame" dmtype="coords:SpaceFrame">
        <ATTRIBUTE dmrole="coords:SpaceFrame.spaceRefFrame" dmtype="ivoa:string" value="ICRS" />
        <INSTANCE dmrole="coords:SpaceFrame.refPosition" dmtype="coords:CustomRefLocation">
          <ATTRIBUTE dmrole="coords:CustomRefLocation.epoch" dmtype="coords:Epoch" value="2015.0"/>
        </INSTANCE>
      </INSTANCE>
    </INSTANCE>
  </GLOBALS>

  <TEMPLATES tableref="Results">
    <INSTANCE dmrole="" dmtype="meas:Position">
      <ATTRIBUTE dmrole="meas:Measure.ucd" dmtype="ivoa:string" value="pos" />
      <INSTANCE dmrole="meas:Measure.coord" dmtype="coords:LonLatPoint">
        <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole="coords:LonLatPoint.lon" ref="" value="" unit="" />
        <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole="coords:LonLatPoint.lat" ref="" value="" unit="" />
        <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole="coords:LonLatPoint.dist" ref="" value="" unit="" />
        <REFERENCE dmrole="coords:Coordinate.coordSys" dmref="SpaceFrame_ICRS" />
      </INSTANCE>
    </INSTANCE>
  </TEMPLATES>
```

# Connect the position fields with the table col.

```
pos_id.mapColumn({"lon": "ra", "lat": "dec", "distance": "parallax"})
```

```
<VODML xmlns:dm-mapping="http://www.ivoa.net/xml/merged-syntax">
  <REPORT status="OK">mapped for Interop Bologna</REPORT>

  <MODEL name="meas" url="https://www.ivoa.net/xml/Meas/20200908/Meas-v1.0.vo-dml.xml" />
  <MODEL name="coords" url="https://www.ivoa.net/xml/STC/20200908/Coords-v1.0.vo-dml.xml" />
  <MODEL name="ivoa" url="https://www.ivoa.net/xml/VODML/IVOA-v1.vo-dml.xml" />

  <GLOBALS>
    <INSTANCE dmid="SpaceFrame_ICRS" dmtype="coords:SpaceSys">
      <INSTANCE dmrole="coords:PhysicalCoordSys.frame" dmtype="coords:SpaceFrame">
        <ATTRIBUTE dmrole="coords:SpaceFrame.spaceRefFrame" dmtype="ivoa:string" value="ICRS" />
        <INSTANCE dmrole="coords:SpaceFrame.refPosition" dmtype="coords:CustomRefLocation">
          <ATTRIBUTE dmrole="coords:CustomRefLocation.epoch" dmtype="coords:Epoch" value="2015.0"/>
        </INSTANCE>
      </INSTANCE>
    </INSTANCE>
  </GLOBALS>

  <TEMPLATES tableref="Results">
    <INSTANCE id="_LonLat_1" dmrole="" dmtype="meas:Position">
      <ATTRIBUTE dmrole="meas:Measure.ucd" dmtype="ivoa:string" value="pos" />
      <INSTANCE dmrole="meas:Measure.coord" dmtype="coords:LonLatPoint">
        <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole="coords:LonLatPoint.lon" ref="ra" unit="deg"/>
        <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole="coords:LonLatPoint.lat" ref="dec" unit="deg"/>
        <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole="coords:LonLatPoint.dist" ref="parallax" unit="mas"/>
        <REFERENCE dmrole="coords:Coordinate.coordSys" dmref="SpaceFrame_ICRS" />
      </INSTANCE>
    </INSTANCE>
  </TEMPLATES>
```

# Add an error to the position

```
error_id = pos_id.addError("Asymmetrical3D")
```

```
<TEMPLATES tableref="Results">

  <INSTANCE id="_LonLat_1" dmrole="" dmtpe="meas:Position">
    <ATTRIBUTE dmrole="meas:Measure.ucd" dmtpe="ivoa:string" value="pos" />
    <INSTANCE dmrole="meas:Measure.coord" dmtpe="coords:LonLatPoint">
      <ATTRIBUTE dmtpe="ivoa:RealQuantity" dmrole="coords:LonLatPoint.lon" ref="ra" unit="deg" />
      <ATTRIBUTE dmtpe="ivoa:RealQuantity" dmrole="coords:LonLatPoint.lat" ref="dec" unit="deg" />
      <ATTRIBUTE dmtpe="ivoa:RealQuantity" dmrole="coords:LonLatPoint.dist" ref="parallax" unit="mas" />
      <REFERENCE dmrole="coords:Coordinate.coordSys" dmref="SpaceFrame_ICRS" />
    </INSTANCE>

    <INSTANCE id="_LonLat_1_Err" dmrole="meas:Measure.error" dmtpe="meas:Asymmetrical3D">
      <COLLECTION dmrole="meas:Asymmetrical3D.plus">
        <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="" value="" unit="" />
        <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="" value="" unit="" />
        <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="" value="" unit="" />
      </COLLECTION>
      <COLLECTION dmrole="meas:Asymmetrical3D.minus">
        <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="" value="" unit="" />
        <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="" value="" unit="" />
        <ATTRIBUTE dmtpe="ivoa:RealQuantity" ref="" value="" unit="" />
      </COLLECTION>
    </INSTANCE>

```

# Connect the error fields with the table columns

```
error_id.mapColumn({"plus": ["ra_error", "dec_error", "parallax_error"]  
                  "minus": ["ra_error", "dec_error", "parallax_error"]})
```

```
<INSTANCE dmrole="" dmtype="meas:Position">  
  <ATTRIBUTE dmrole="meas:Measure.ucd" dmtype="ivoa:string" value="pos" />  
  <INSTANCE dmrole="meas:Measure.coord" dmtype="coords:LonLatPoint">  
    <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole="coords:LonLatPoint.lon" ref="" />  
    <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole="coords:LonLatPoint.lat" ref="" />  
    <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole="coords:LonLatPoint.dist" ref="" />  
    <REFERENCE dmrole="coords:Coordinate.coordSys" dmref="SpaceFrame_ICRS" />  
  </INSTANCE>  
  
  <INSTANCE dmrole="meas:Measure.error" dmtype="meas:Asymmetrical3D">  
    <COLLECTION dmrole="meas:Asymmetrical3D.plus">  
      <ATTRIBUTE dmtype="ivoa:RealQuantity" ref="ra_error" unit="mas"/>  
      <ATTRIBUTE dmtype="ivoa:RealQuantity" ref="dec_error" unit="mas"/>  
      <ATTRIBUTE dmtype="ivoa:RealQuantity" ref="parallax_error" unit="mas"/>  
    </COLLECTION>  
    <COLLECTION dmrole="meas:Asymmetrical3D.minus">  
      <ATTRIBUTE dmtype="ivoa:RealQuantity" ref="ra_error" unit="mas"/>  
      <ATTRIBUTE dmtype="ivoa:RealQuantity" ref="dec_error" unit="mas"/>  
      <ATTRIBUTE dmtype="ivoa:RealQuantity" ref="parallax_error" unit="mas"/>  
    </COLLECTION>  
  </INSTANCE>  
</INSTANCE>
```

# Validator

Annotated VOTABLE

MIVOT instance (mapped)

```
<INSTANCE dmrole="meas:Measure.coord" dmtype
="coords:LonLatPoint">
  <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole
="coords:LonLatPoint.lon" value="132.56"
unit="deg"/>
  <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole
="coords:LonLatPoint.lat" value="-45.87"
unit="deg"/>
  <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole
="coords:LonLatPoint.dist" value="0.17"
unit="mas"/>
</INSTANCE>
```

**CHECKING:**

- All the elements of the **mapped instance** must be in the generic instance
- **the reverse can be false**

VODML File

MIVOT instance (generic)

```
<INSTANCE dmrole="meas:Measure.coord" dmtype
="coords:LonLatPoint">
  <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole
="coords:LonLatPoint.lon" value="" ref=""
unit=""/>
  <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole
="coords:LonLatPoint.lat" value="" ref=""
unit=""/>
  <ATTRIBUTE dmtype="ivoa:RealQuantity" dmrole
="coords:LonLatPoint.dist" value="" ref=""
unit=""/>
</INSTANCE>
```

<https://github.com/ivoa/mivot-validator>

# Use the validator feature to generate MIVOT snippets

```
def main():
    """
    Package launcher (script)
    """
    if len(sys.argv) != 4:
        print("USAGE: mivot-snippet-gen [class_name] [output_dir] [output_name]")
        print("    Create MIVOT snippets from VODML files")
        print("    :class_name: either a simple file or a class name")
        print("    :output_dir: path to the output directory")
        print("    :output_name: name of the output file")
        print("    exit status: 0 in case of success, 1 otherwise")
        sys.exit(1)
```

# MIVOT snippet generation

Command issued from  
the validator package

Selection of one concrete class

```
$ mivot-snippet-gen "coords.SpaceFrame" "RefLocation=CustomSpaceLocation"
```

Class to be generated

<INSTANCE dmrole="dmtype="coords:SpaceFrame">

Requested class

<!-- A SpaceFrame is specified by its reference frame (orientation), and a reference position (origin).  
Currently only standard reference frames are allowed. An equinox MUST be provided for pre-ICRS reference frames.  
-->

<INSTANCE dmrole="coords:SpaceFrame.refPosition" dmtype="coords:CustomRefLocation">

Requested concrete class

<!-- A custom reference location in phase space (position and velocity).  
Position and velocity are given as coordinates with an associated SpaceFrame.  
An epoch MAY be provided to further refine the location." -->  
<!-- We define epoch as a primitive data type with the expected form "<type><year>"  
where type = "J" or "B" for Julian or Besselian respectively -->  
<ATTRIBUTE dmrole="coords:CustomRefLocation.epoch" dmtype="coords:Epoch" ref="@@@@" value=""/>  
<INSTANCE dmrole="coords:CustomRefLocation.position" dmtype="coords:LonLatPoint">

<!-- A spatial coordinate in a Spherical coordinate space defining a Celestial position in Latitude and Longitude.  
Any associated CoordSpace MUST conform to this description. If no CoordSpace is provided,  
a Standard Spherical CoordSpace is assumed. Values for unused/undefined dimensions need not be provided.-->

VODML description of the element

<ATTRIBUTE dmrole="coords:LonLatPoint.lon" dmtype="ivoa:Quantity"  
<ATTRIBUTE dmrole="coords:LonLatPoint.lat" dmtype="ivoa:Quantity"  
<ATTRIBUTE dmrole="coords:LonLatPoint.dist" dmtype="ivoa:Quantity" unit="" ref="@@@@" value=""/>  
</INSTANCE>

<INSTANCE dmrole="coords:CustomRefLocation.velocity" dmtype="coords:LonLatPoint">  
<!-- A spatial coordinate in a Spherical coordinate space defining a Celestial position in Latitude and Longitude.  
Any associated CoordSpace MUST conform to this description. If no CoordSpace is provided, a Standard Spherical  
CoordSpace is assumed. Values for unused/undefined dimensions need not be provided. -->  
<ATTRIBUTE dmrole="coords:LonLatPoint.lon" dmtype="ivoa:Quantity" unit="" ref="@@@@" value=""/>  
<ATTRIBUTE dmrole="coords:LonLatPoint.lat" dmtype="ivoa:Quantity" unit="" ref="@@@@" value=""/>  
<ATTRIBUTE dmrole="coords:LonLatPoint.dist" dmtype="ivoa:Quantity" unit="" ref="@@@@" value=""/>  
</INSTANCE>

</INSTANCE>  
<ATTRIBUTE dmrole="coords:SpaceFrame.spaceRefFrame" dmtype="ivoa:string" ref="@@@@" value=""/>  
<!-- We define epoch as a primitive data type with the expected form "<type><year>" where type = "J" or "B"  
for Julian or Besselian respectively, and year is expressed as a decimal year. e.g.: "B1950", "J2000.0" -->  
<ATTRIBUTE dmrole="coords:SpaceFrame.equinox" dmtype="coords:Epoch" ref="@@@@" value=""/>  
<ATTRIBUTE dmrole="coords:SpaceFrame.planetaryEphem" dmtype="ivoa:string" ref="@@@@" value=""/>  
</INSTANCE>



# Server Side Tooling (TAP service)

- **The principle**

- Analyze the query to see which columns can be mapped on which classes
- For each mappable class: add the corresponding XML block in the MIVOT BLOCK of the VOTABLE

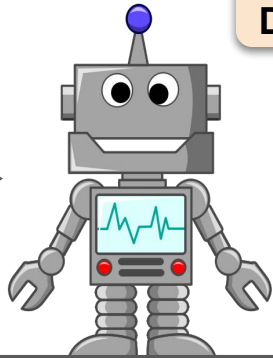
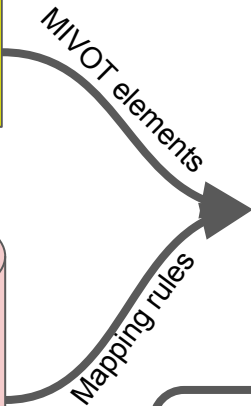
- **Achievements**

- ADASS 2021 BoF, "TAP and the Data Models"
- XTapDB service working for two XMM-Newton tables

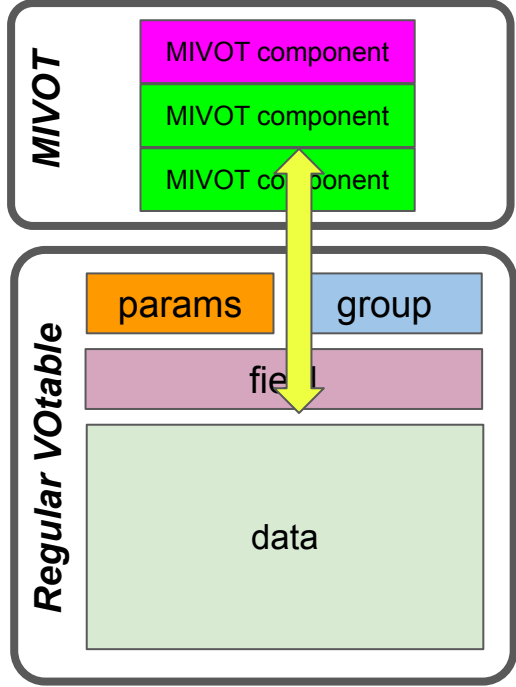
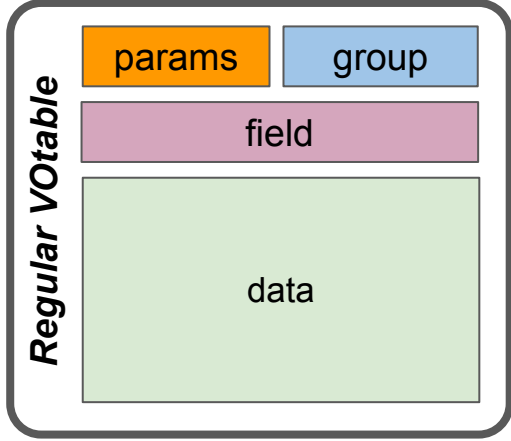
# Server Side Tooling (TAP service add-on)



TAP Schema **extended** with tables giving the **mapping rules** for individual tables



The mapper post-processes the result table  
Data is mapped on the MANGO model



# Server Side Tooling (TAP service add-on)

- **Build a VOLLT add-on**
  - Enable any TAP service to map query results on MANGO
  - Mapping rules in either a TAP schema extension or in other specific resources.
  - <https://github.com/lmichel/vollt-mivot-extension> (empty but the Wiki)
- **Working with Gilles (Vizier) to assess a MIVOT implementation in Vizier**
  - Anyone is welcome to join



# Documentation

- **All implementations are documented on the Wiki pages of their repositories.**
  - Need for a portal, e.g. on <https://github.com/ivoa-std/ModelInstanceInVot>/README
- **VOExamples revival**
  - Web resource developed for ObsCore (not continued)
  - <https://saada.unistra.fr/voexamples>
  - A MIVOT section is being added (M. Louys, J. Abid)



<b>Module</b>	<b>status</b>	<b>GitHub availability</b>
Python parser	Working + all MIVOT features supported	yes
AstroPyVo client	Prototype + Running meetings	yes
Snippet builder	Working along with the validator	yes (not merged yet)
AstroPyVo server	Concept + ongoing discussions	
VOLLT add-on	Prototype (XMM)	Online Private
documentation	Being worked out	Yes (not merged)

**We couldn't complete everything without help !**

# Conclusions

- **We have prototypes for every part of the MIVOT workflow....**
  - We have shown that this workflow is workable and useful
  - It can also be used to test new models
- **... that must be finalized and documented**
  - AstroPyVO PR
  - VOLLT extension
  - Other...
- **We can't do everything on our own**
  - Need active contributors
  - Need testers
  - Need modelers



# Models and VOTables

~ Part II ~

MIVOT

L. Michel



# Connect Data with Models

- **Data annotation with UTypes**

- Group items refer to model leaves
- Data response kinked to models by a key mechanism



works well

- **Mapping data on models in query responses**

- Data response comes with a whole description of the model they refer to.
- The client has enough material to build model instance from the data.



Stepping out of the rut ?

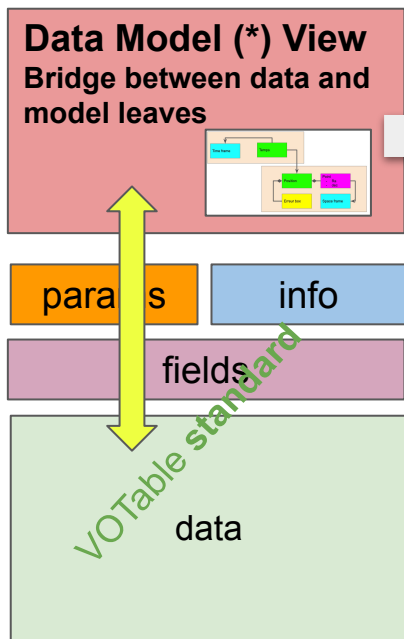
## 2 Purposes

- **Map data on VO models**
- **Test ongoing models on reel data**

# **Connect Data with Models: The Baseline**

- **2017, the community agreed on the annotation baselines**
  - Define a syntax for connecting data and models in VOTables
  - This syntax had to be arranged in a way that does not alter the schema of VOTables
  - Must be in XML
  - Must obey a schema independent of any particular model
- **2023, Model Instance in VOTable (MIVOT) voted by the Exec this Spring**
  - **Strong TCG requirement for guiding implementers**
    - Explain the process
    - Provide educational materials
    - Provide code
    - Complete the model for Archival Data (MANGO)
  -

# MIVOT: Associate a model view with tabular data



- **Describe how data are connected to each other**
  - Association flag/error/measure
- **Improve semantic description**
  - Add semantic to data associations
- **Add possibly missing meta-data**
  - Filter descriptions
  - Flag values
- **View not depending on the data provider**

(\*) Let's assume we have data models serialized in VODML

# MIVOT: Mapping Block Location

- The model mapping is embedded in one XML Block
  - Located in a `resource[@type=meta]`
  - Scope limited to the parent `resource[@type=results]`
  - Mapped data can be distributed over multiple tables within that `resource`

```
<?xml version="1.0" encoding="UTF-8"?>
<VOTABLE xmlns="http://www.ivoa.net/xml/VOTable/v1.3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="1.3">
  <RESOURCE type="results">
    <RESOURCE type="meta">
      <VODML xmlns:dm-mapping="http://www.ivoa.net/xml/merged-syntax">
    </RESOURCE>
    <TABLE name="Results">
      <PARAM ID="_title" name="title" value="TilteReadInParam" datatype="char" arraysize="*" />
      <FIELD ID="_poserr_148" name="oidsaada" datatype="long" ucd="meta.id;meta.main" />
      <DATA>
    </TABLE>
    <TABLE name="OtherResults">
    <TABLE name="Spectra">
  </RESOURCE>
</VOTABLE>
```

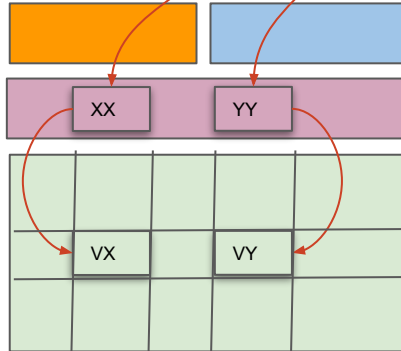
# MIVOT: Provide Model Views to Users

```
<VODML>
  <GLOBALS>
    <....>
    <INSTANCE
      dmid= spaceframe
    </GLOBALS>
  <TEMPLATES>
    <INSTANCE>
      ... dhref= spaceframe
      ... ref=xx ref=yy...
    </INSTANCE>
  </TEMPLATES>
</VODML>
```



**Model View**

**USER**



**XML Serialization of the model view of the data rows**

- References resolved
- Components gathered

# MIVOT: Example

```
<TEMPLATES tableref="Spectra">
  <INSTANCE dmttype="test:Spectrum">
    <ATTRIBUTE dmrole="test:detection.num" dmttype="ivoa:string?" ref=" _spc_148" />
    <ATTRIBUTE dmrole="test:detection.id" dmttype="ivoa:real" ref=" _foreign_spectra" />
  </INSTANCE>
</TEMPLATES>
</VODML>
RESOURCE>
TABLE name="Spectra">
  <FIELD ID=" _spc_148" name="oidsaada" datatype="char" arraysize="*" ucd="meta.id;meta.main" />
  <FIELD ID=" _foreign_spectra" name="cle" datatype="long" ucd="meta.id;meta.main" />
  <DATA>
    <TABLEDATA>
      <TR>
        <TD>Spectrum 11</TD>
        <TD>1</TD>
      </TR>
      <TR>
        <TD>Spectrum 12</TD>
        <TD>1</TD>
      </TR>
    </TABLEDATA>
  </TABLE>
</RESOURCE>
```

**TEMPLATE** [ @tableref=Spectra ] maps the data of **TABLE** [ @name=Spectra ]

Each row of that table can be interpreted an instance of type **test:Spectrum**

The connection between model leaves and table columns is achieved by the **ATTRIBUTE** [ @ref ] = **FIELD** [ @ID ] condition

# Syntax Overview

- **Mapping block structure**

- One <VODML> block, the container
- Followed by one <REPORT> optional block giving the status of the mapping process
- Followed but one <MODEL> element per used model
- Followed by one <GLOBALS> element
  - Mapping of quantities that do not relate to table rows

```
<VODML xmlns="http://www.ivoa.net/xml/merged-syntax" >
  <REPORT status="OK">Hand made mapping</REPORT>
  <MODEL name="mango" url="file:/Users/sao/Documents/IVOA/GitHub/ivoa-dm-examples/tmp/Mango-v1.0.vo-dm
  <MODEL name="cube" url="https://volute.g-vo.org/svn/trunk/projects/dm/Cube/vo-dml/Cube-1.0.vo-dml.x
  <MODEL name="ds" url="https://volute.g-vo.org/svn/trunk/projects/dm/DatasetMetadata/vo-dml/Datase
  <MODEL name="meas" url="https://www.ivoa.net/xml/Meas/20200908/Meas-v1.0.vo-dml.xml" />
  <MODEL name="coords" url="https://www.ivoa.net/xml/STC/20200908/Coords-v1.0.vo-dml.xml" />
  <MODEL name="ivoa" url="https://www.ivoa.net/xml/VODML/IVOA-v1.vo-dml.xml" />
  <GLOBALS>
  <TEMPLATES tableref="_PKTable">
  <TEMPLATES tableref="Results">
</VODML>
```



# Mapping table rows with TEMPLATES

```
<TEMPLATES tableref="Spectra">
  <INSTANCE dmttype="test:Spectrum">
    <ATTRIBUTE dmrole="test:detection.num" dmttype="ivoa:string?" ref="_spc_148" />
    <ATTRIBUTE dmrole="test:detection.id" dmttype="ivoa:real" ref="_foreign_spectra" />
  </INSTANCE>
</TEMPLATES>
</VODML>
RESOURCE>
TABLE name="Spectra">
  <FIELD ID="_spc_148" name="oidsaada" datatype="char" arraysize="*" ucd="meta.id;meta.main" />
  <FIELD ID="_foreign_spectra" name="cle" datatype="long" ucd="meta.id;meta.main" />
  <DATA>
    <TABLEDATA>
      <TR>
        <TD>Spectrum 11</TD>
        <TD>1</TD>
      </TR>
      <TR>
        <TD>Spectrum 12</TD>
        <TD>1</TD>
      </TR>
    </TABLEDATA>
  </TABLE>
</RESOURCE>
```

**TEMPLATE**[@tableref=Spectra] maps the data of **TABLE**[@name=Spectra]

Each row of that table can be interpreted an instance of type **test:Spectrum**

The connection between model leaves and table columns is achieved by the **ATTRIBUTE**[@ref]=**FIELD**[@ID] condition

# Using References

```
<INSTANCE dmtpe="sed:Sed">
  <INSTANCE dmtpe="sed:Source" dmrole="sed:Sed.source">
    <ATTRIBUTE dmtpe="ivoa:string" dmrole="sed:Source.name"
    value="..." />
  </INSTANCE>
  <INSTANCE dmtpe="sed:Position" dmrole="sed:Source.position">
    <ATTRIBUTE dmtpe="ivoa:real" dmrole="sed:Position.longitude" ref="ra" />
    <ATTRIBUTE dmtpe="ivoa:real" dmrole="sed:Position.latitude" ref="dec" />
    <ATTRIBUTE dmtpe="ivoa:vounit" dmrole="sed:Position.unit" value="deg" />
  </INSTANCE>
  <COLLECTION dmrole="sed:Sed.points">
    <INSTANCE dmtpe="sed:PhotPoint">
      <ATTRIBUTE dmtpe="ivoa:real" dmrole="sed:PhotPoint.flux" ref="flux"
      unit="erg/s/cm^2" />
      <REFERENCE dmrole="sed:PhotPoint.photcal" dmref="_XMMSL2_EB6" />
    </INSTANCE>
    <INSTANCE dmtpe="sed:PhotPoint">
    <INSTANCE dmtpe="sed:PhotPoint">
    <INSTANCE dmtpe="sed:PhotPoint">
    <INSTANCE dmtpe="sed:PhotPoint">
    <INSTANCE dmtpe="sed:PhotPoint">
    <INSTANCE dmtpe="sed:PhotPoint">
    <INSTANCE dmtpe="sed:PhotPoint">
    <INSTANCE dmtpe="sed:PhotPoint">
    <INSTANCE dmtpe="sed:PhotPoint">
    <INSTANCE dmtpe="sed:PhotPoint">
    </COLLECTION>
  </INSTANCE>
```

Each table row contains one SED  
*Toy model*

The connection between photometric points and calibration is achieved by the **REFERENCE** [ @dmref ] = **INSTANCE** [ @dmid ] condition

```
<GLOBALS>
  <INSTANCE dmtpe="photdm:PhotCal" dmid="_XMMSL2_EB6">
  <INSTANCE dmtpe="photdm:PhotCal" dmid="_XMMSL2_EB7">
  <INSTANCE dmtpe="photdm:PhotCal" dmid="_XMMSL2_EB8">
  <INSTANCE dmtpe="photdm:PhotCal" dmid="_2MASS_H">
  <INSTANCE dmtpe="photdm:PhotCal" dmid="_2MASS_J">
  <INSTANCE dmtpe="photdm:PhotCal" dmid="_2MASS_Ks">
  <INSTANCE dmtpe="photdm:PhotCal" dmid="_WISE_W1">
  <INSTANCE dmtpe="photdm:PhotCal" dmid="_WISE_W2">
  <INSTANCE dmtpe="photdm:PhotCal" dmid="_WISE_W3">
  <INSTANCE dmtpe="photdm:PhotCal" dmid="_WISE_W4">
  <INSTANCE dmtpe="photdm:PhotCal" dmid="_GAIA2_G">
</GLOBALS>
```

# A Very Complex Object

```
<GLOBALS>
<INSTANCE dmtpe="photdm:PhotCal" dmid="_XMMSL2_EB8">
<INSTANCE dmtpe="photdm:PhotCal" dmid="_XMMSL2_EB7">
<INSTANCE dmtpe="photdm:PhotCal" dmid="_XMMSL2_EB6">
<INSTANCE dmtpe="photdm:PhotCal" dmid="_2MASS_H">
<INSTANCE dmtpe="photdm:PhotCal" dmid="_2MASS_J">
<INSTANCE dmtpe="photdm:PhotCal" dmid="_2MASS_Ks">
<INSTANCE dmtpe="photdm:PhotCal" dmid="_WISE_W1">
<INSTANCE dmtpe="photdm:PhotCal" dmid="_WISE_W2">
<INSTANCE dmtpe="photdm:PhotCal" dmid="_WISE_W3">
<INSTANCE dmtpe="photdm:PhotCal" dmid="_WISE_W4">
<INSTANCE dmtpe="photdm:PhotCal" dmid="_GAIA2_G">
</GLOBALS>
```

```
<INSTANCE dmtpe="photdm:PhotCal" dmid="_2MASS_H">
<ATTRIBUTE dmrole="photdm:PhotCal.identifier" dmtpe="ivoa:string"
value="_2MASS/2MASS.H/Vega" />
<!-- Magnitude System -->
<INSTANCE dmtpe="photdm:MagnitudeSystem" dmrole="photdm:PhotCal.magnitudeSystem">
<ATTRIBUTE dmrole="photdm:MagnitudeSystem.type" dmtpe="photdm:TypeOfMagSystem"
value="Vega" />
<ATTRIBUTE dmrole="photdm:MagnitudeSystem.referenceSpectrum" dmtpe="ivoa:anyURI"
value="http://svo2.cab.inta-csic.es/theory/fps/morefiles/vega.dat" />
</INSTANCE>
<!-- Zero Point -->
<INSTANCE dmtpe="photdm:PogsonZeroPoint" dmrole="photdm:PhotCal.zeroPoint">
<ATTRIBUTE dmrole="photdm:ZeroPoint.type" dmtpe="ivoa:string" value="Pogson" />
<ATTRIBUTE dmrole="photdm:ZeroPoint.referenceMagnitudeValue" dmtpe="ivoa:real" value="0" />
<INSTANCE dmtpe="photdm:Flux" dmrole="photdm:ZeroPoint.flux">
<ATTRIBUTE dmrole="photdm:Flux.ucd" dmtpe="ivoa:UCD" value="phot.flux;meta.modelled" />
<ATTRIBUTE dmrole="photdm:Flux.unitexpression" dmtpe="ivoa:Unit" value="Jy" />
<ATTRIBUTE dmrole="photdm:Flux.value" dmtpe="ivoa:real" value="1024.0" />
</INSTANCE>
</INSTANCE>
<!-- Filter -->
<INSTANCE dmtpe="photdm:PhotometryFilter" dmrole="photdm:PhotCal.photometryFilter">
<ATTRIBUTE dmrole="photdm:PhotometryFilter.fpsidentifier" dmtpe="ivoa:string"
value="ivo://svo/fps" />
<ATTRIBUTE dmrole="photdm:PhotometryFilter.identifier" dmtpe="ivoa:string"
value="_2MASS/2MASS.H" />
<ATTRIBUTE dmrole="photdm:PhotometryFilter.name" dmtpe="ivoa:string" value="2MASS H" />
<ATTRIBUTE dmrole="photdm:PhotometryFilter.description" dmtpe="ivoa:string"
value="2MASS H" />
<ATTRIBUTE dmrole="photdm:PhotometryFilter.bandname" dmtpe="ivoa:string" value="H" />
<!-- Spectral Location -->
<INSTANCE dmtpe="photdm:SpectralLocation"
dmrole="photdm:PhotometryFilter.spectralLocation">
<ATTRIBUTE dmrole="photdm:SpectralLocation.ucd" dmtpe="ivoa:UCD"
value="em.wl.effective" />
<ATTRIBUTE dmrole="photdm:SpectralLocation.unitexpression" dmtpe="ivoa:Unit"
value="Angstrom" />
<ATTRIBUTE dmrole="photdm:SpectralLocation.value" dmtpe="ivoa:real" value="16620.0" />
</INSTANCE>
<!-- Band width -->
<INSTANCE dmtpe="photdm:Bandwith" dmrole="photdm:PhotometryFilter.spectralLocation">
<ATTRIBUTE dmrole="photdm:Bandwith.ucd" dmtpe="ivoa:UCD"
value="instr.bandwidth;stat.fwhm" />
<ATTRIBUTE dmrole="photdm:Bandwith.unitexpression" dmtpe="ivoa:Unit" value="Angstrom" />
<ATTRIBUTE dmrole="photdm:Bandwith.extent" dmtpe="ivoa:real" value="2509.4034987068" />
<ATTRIBUTE dmrole="photdm:Bandwith.start" dmtpe="ivoa:real" value="14787.378640179" />
<ATTRIBUTE dmrole="photdm:Bandwith.stop" dmtpe="ivoa:real" value="18231.020407164" />
</INSTANCE>
<!-- Transmission Curve -->
<INSTANCE dmtpe="photdm:TransmissionCurve" dmrole="photdm:Bandwith.transmissionCurve">
<INSTANCE dmtpe="photdm:Access" dmrole="photdm:TransmissionCurve.access">
<ATTRIBUTE dmrole="photdm:Access.reference" dmtpe="ivoa:anyURI"
value="http://svo2.cab.inta-csic.es/theory/fps/fps.php?PhotCalID=2MASS/2MASS.H/Vega" />
<ATTRIBUTE dmrole="photdm:Access.size" dmtpe="ivoa:integer" value="-1" />
<ATTRIBUTE dmrole="photdm:Access.format" dmtpe="ivoa:string" value="VOTable" />
</INSTANCE>
</INSTANCE>
</INSTANCE>
```

# Table Join (advanced feature)

```
<TEMPLATES tableref="_PKTable">
  <INSTANCE dmid="_TimeSeries" dmrole="" dmtype="cube:SparseCube">
    <REFERENCE dmrole="cube:DataProduct.dataset" sourceref="_Datasets">
      <FOREIGN_KEY ref="_pksrcid"/>
    </REFERENCE>
    <COLLECTION dmrole="cube:SparseCube.data">
      <JOIN dmref="_ts_data">
        <WHERE foreignkey="_srcid" primarykey="_pksrcid" />
        <WHERE foreignkey="_band" primarykey="_pkband" />
      </JOIN>
    </COLLECTION>
  </INSTANCE>
</TEMPLATES>
<TEMPLATES tableref="Results">
  <INSTANCE dmid="_ts_data" dmrole="" dmtype="cube:NDPoint">
    <COLLECTION dmrole="cube:NDPoint.observable">
      <INSTANCE dmtype="cube:Observable">
        <ATTRIBUTE dmrole="cube:DataAxis.dependent" dmtype="ivoa:boolean" value="False"/>
      </INSTANCE>
    </COLLECTION>
  </INSTANCE>
</TEMPLATES>
```

The `cube:SparseCube.data` is populated with joined data

- That are instances of the table `Results[@dmid=_ts_data]`
- That use both columns `_band` and `_srcid` of table `Results` as foreign keys
- And both columns `_pkband` and `_pksrcid` of table `PKTable` as primary keys

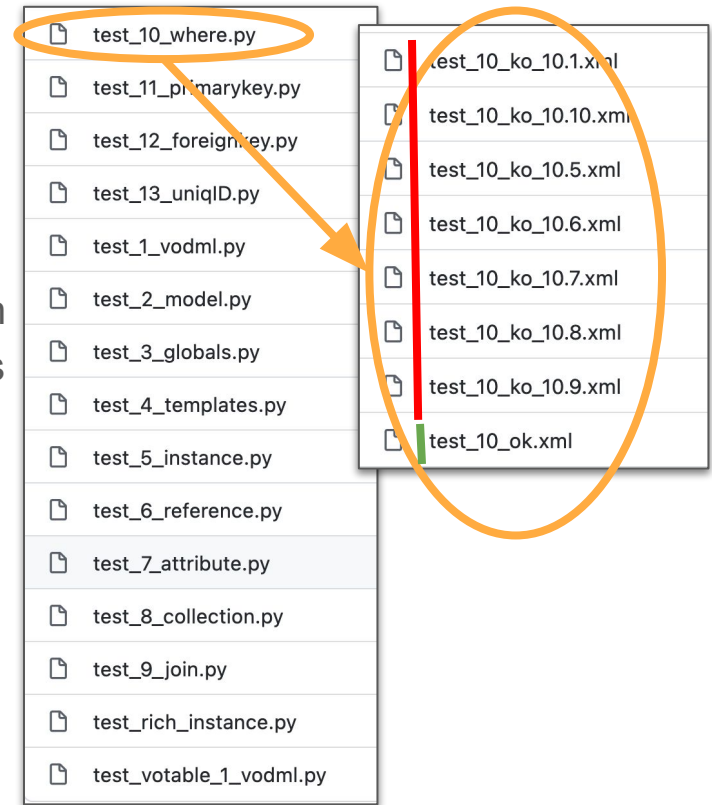
# MIVOT: Design validation

- **Based on many unit tests**

- One test suite per element
  - One snippet file with all good patterns
  - Some failing snippets
    - Verify that tests fail for the expected reason
- Snippets can be used as a library of standard patterns
  - Discover the syntax (reviewers could thank us)
  - Exercise the annotation

- **Validator**

- All rules are enforced by both MIVOT XSD schema and VODML files



# MIVOT: the code repository

- **Most of the implementations gathered in one repository**
  - <https://github.com/ivoa/modelinstanceinvot-code>
    - Parser
    - data sample
    - test programs
    - notebooks
  - Let's admit that this is a little bit catch-all

# MIVOT: Status

- **REC now**

- RFC page

<https://wiki.ivoa.net/twiki/bin/view/IVOA/DataAnnotation>

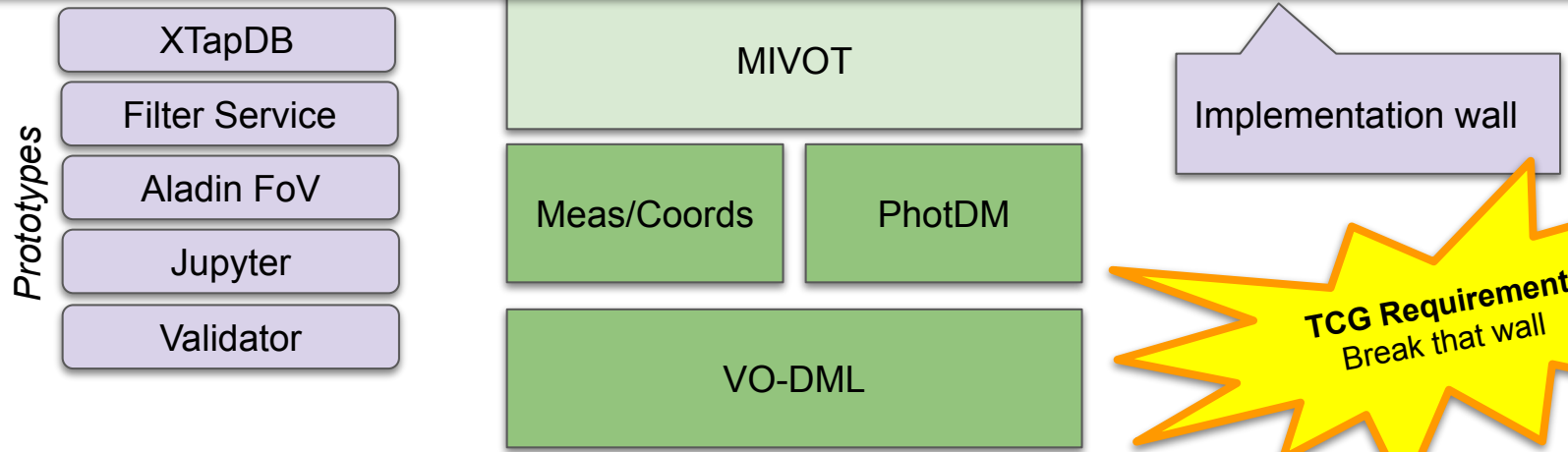
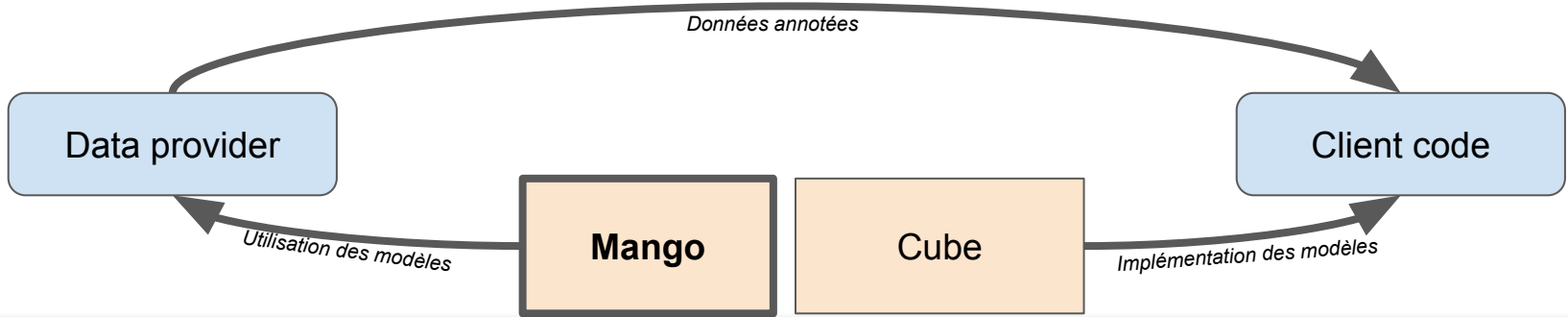
- GitHub repository

<https://github.com/ivoa-std/ModelInstanceInVot>

- **Rather (self) Satisfied**

- Smooth VOTable integration
- Simple annotations for the simple cases
- Support all use-cases proposed in the DM workshop
- Working validator
- Voted by the Exec

# Today's situation: making MIVOT to be used





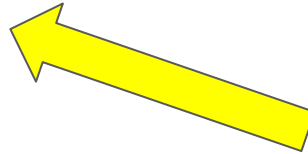
# Next Steps

- **Models with added value**

- **Models for scientific products**

- Cube (MCD): flexible model for complex data such as images or event lists.
- Uses components imported from other models (Measure/Coords)

- **Model for archival data (MANGO)**



- **Implementation Toolbox**

- Documentation
- Examples
- Tools

# Models and VOTables

~ Part III ~

## MANGO

L. Michel

# Define a model for archival data

## Strategy for defining use-cases

**2018:** Poll to get requirements for parameters describing sources

- A few answers
- The Gaia case among them

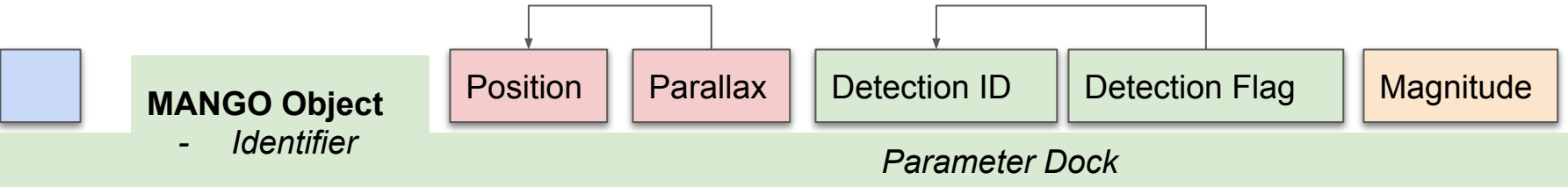
**2019:** Source DM session in Paris where various people including scientists was invited to present their requirements

- we got a robust set of scientifically relevant use cases
- We got some developers and provider requirements

# A model for archival data

- **Identification of not, poorly or not properly defined properties in VOTABLES**
  - Complex errors
  - Orbital systems
  - Complex forms
  - Interpretation of statuses and labels
  - Association of measures between them
  - Association of measures and flags
  - Define the semantics of the links
  - Calibration status (photometric or other)
  - Attach semantics to measurements
- **Limit as much as possible the list of mandatory fields**
  - Few common points between the different data sets
  - Put only the present measurements
  - Do not duplicate metadata already in VOTables.

# MANGO: Aggregating model components



**Meas/Coords**  
Fine-grained definition of spatial and temporal coordinates

**PhotDM**

- Photometric Calibration
- Filters
- Photometric Systems

**Mango Classes**

- Labels
- Colors
- TBD

**Provenance**

Gather all parameters in a **Dock** to be able to represent the source as a single object and not as a set of independent entities

*rather*



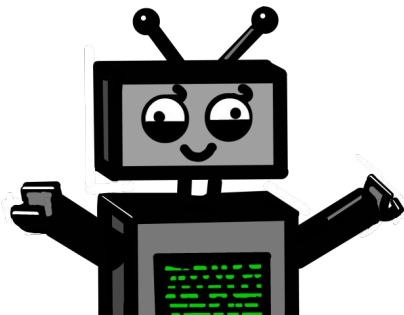
*than*



# Working with MANGO



{meta data}  
 - TAP Schema  
 - Provenance  
 - DataSet



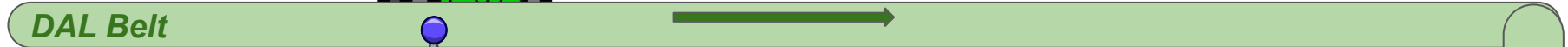
**Component box**

- Dataset
- Position
- Magnitude
- Redshift
- Proper Motion

Sex	Metabolic rate	Sex	Metabolic rate
Male	525.8	Female	727.7
Male	605.7	Female	1086.5
Male	843.3	Female	1091.0
Male	1195.5	Female	1361.3
Male	1945.6	Female	1490.5
Male	2335.6	Female	1956.1
Male	2950.0		

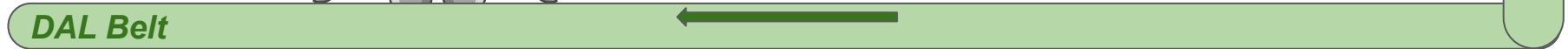
MANGO gives way to pack properties

Sex	Metabolic rate	Sex	Metabolic rate
Male	525.8	Female	727.7
Male	605.7	Female	1086.5
Male	843.3	Female	1091.0
Male	1195.5	Female	1361.3
Male	1945.6	Female	1490.5
Male	2335.6	Female	1956.1
Male	2950.0		

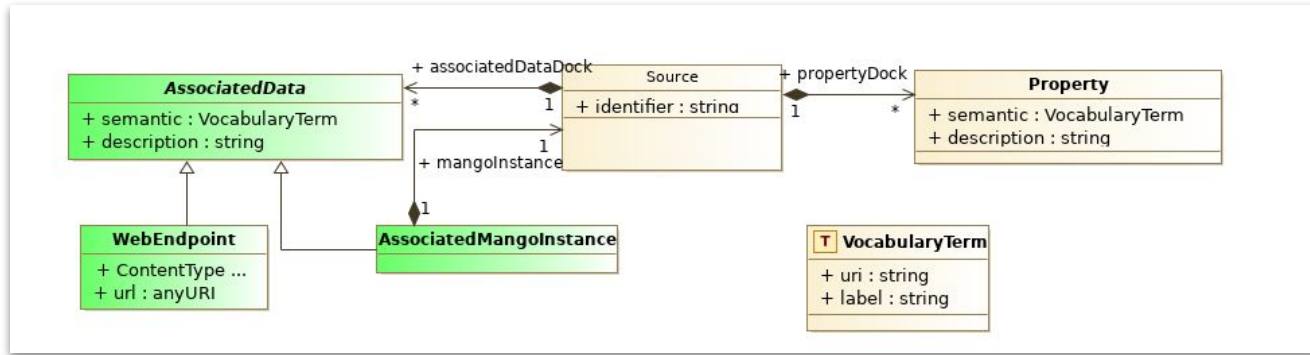


ICRS Position    Magnitude B    Proper Motion    Dataset

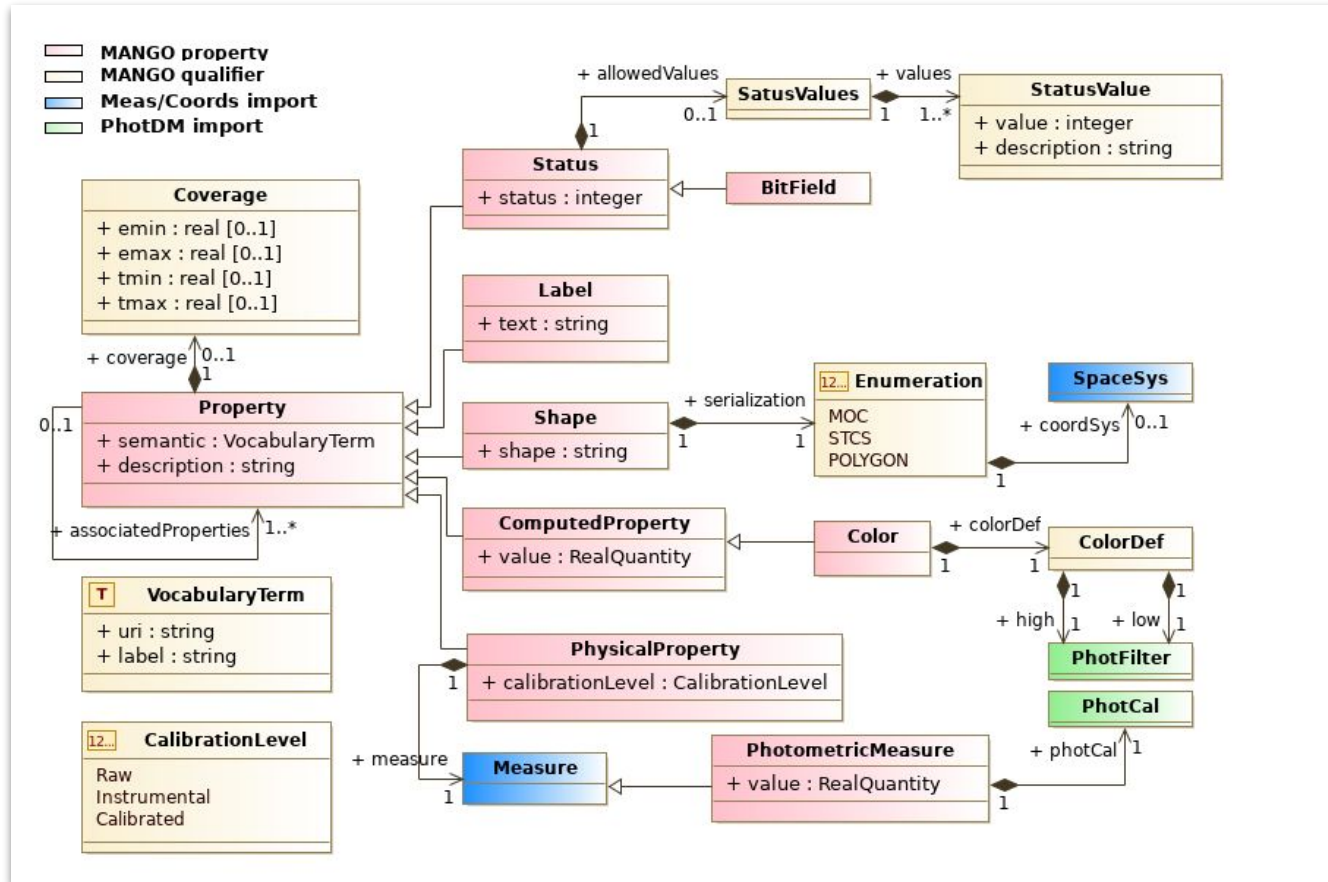
Sex	Metabolic rate	Sex	Metabolic rate
Male	525.8	Female	727.7
Male	605.7	Female	1086.5
Male	843.3	Female	1091.0
Male	1195.5	Female	1361.3
Male	1945.6	Female	1490.5
Male	2335.6	Female	1956.1
Male	2950.0		



# Mango Today: Seen from afar



# Mango Today: Seen from closer





# MANGO: Status

- **MANGO draft revamped in 2022 after the DM workshop.**
  - Many simplifications
  - Project pended due to the MIVOT RFC
- **It is time to contribute**
  - Proposal for specific use-cases
  - Data sample (can be tested with MIVOT)
  - Proposal for specific patterns
  - Example review

Having a stable version of MIVOT with tools and validators allows to exercise models on real data



# Models and VOTables

~ Part IV ~

## Implementation

L. Michel

# Moving toward Operational Implementation

- **Validator**
  - Validate annotated VOTable against the mapped models.
- **Documentation**
  - Help users to figure out what are the steps of the process
- **Tooling for the server side**
  - Showing up a generic implementation for a TAP service
- **Tooling for the client side**
  - Make a PR enabling AstroPyVO to work with annotated data.
- **Register MIVOT enable services**

These tools will help for both **implementing** services and **exercising model proposals** on **real data**.

# Validator

- **Package available on GitHub (Python)**
  - [https://github.com/ivoa/mivot-validator/tree/snippet\\_dev](https://github.com/ivoa/mivot-validator/tree/snippet_dev)
- **Validation against the XSD**
  - VOTable validation against 1.3 XSD
  - VOTable validation against MVOT.xsd (XSD1.1)
  - Validation of the mapped classes against the VODML model
- **A serendipitous effect (snippet\_dev branch)**
  - The validation against model process requires at some point to extract MIVOT snippets from the VODML files.
  - This function is being promoted as a package feature.
  - This is a major help for people working with annotations

# Documentation

- **All implementations are documented on the Wiki pages of their repositories.**
  - Need for a portal, e.g. on <https://github.com/ivoa-std/ModelInstanceInVot>/README
- **VOExamples revival**
  - Web resource developed for ObsCore (not continued)
  - <https://saada.unistra.fr/voexamples>
  - A MIVOT section is being added (M. Louys, J. Abid)

# Server Side Tooling (TAP service)

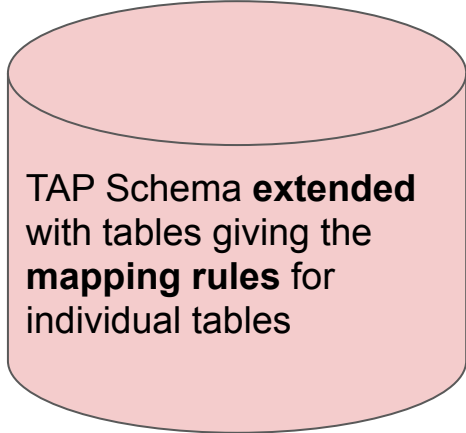
- **The principle**

- Analyze the query to see which columns can be mapped on which classes
- For each mappable class: add the corresponding XML block in the MIVOT BLOCK of the VOTABLE

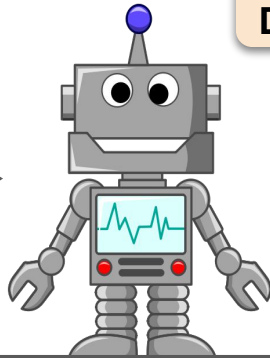
- **Achievements**

- ADASS 2021 BoF, "TAP and the Data Models"
- XTapDB service working for two XMM-Newton tables

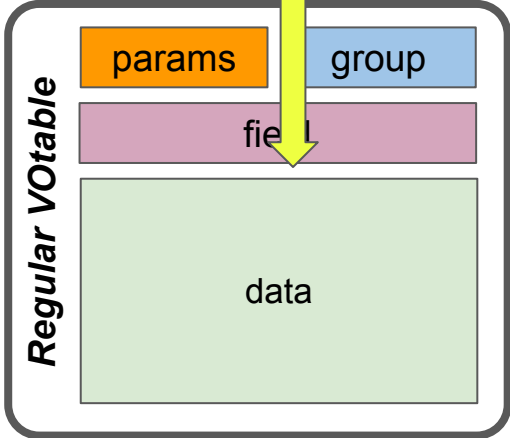
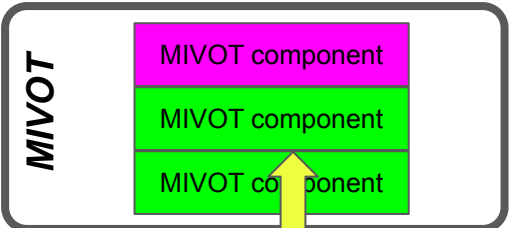
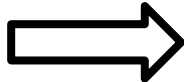
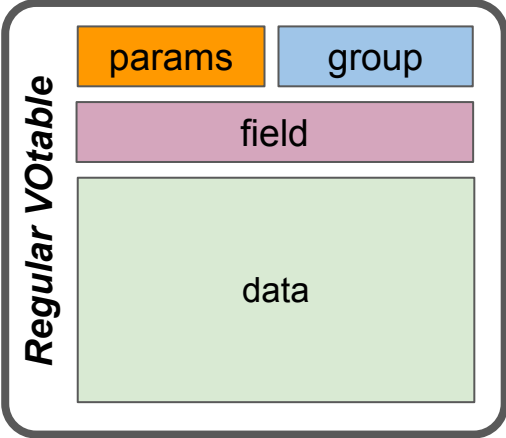
# Server Side Tooling (TAP service add-on)



MIVOT elements



The mapper post-processes the result table  
Data is mapped on the MANGO model



# Server Side Tooling (TAP service add-on)

- **Build a VOLLT add-on**
  - Enable any TAP service to map query results on MANGO
  - Mapping rules in either a TAP schema extension or in other specific resources.
  - <https://github.com/lmichel/vollt-mivot-extension> (empty but the Wiki)
- **Working with Gilles (Vizier) to assess a MIVOT implementation in Vizier**
  - Anyone is welcome to join





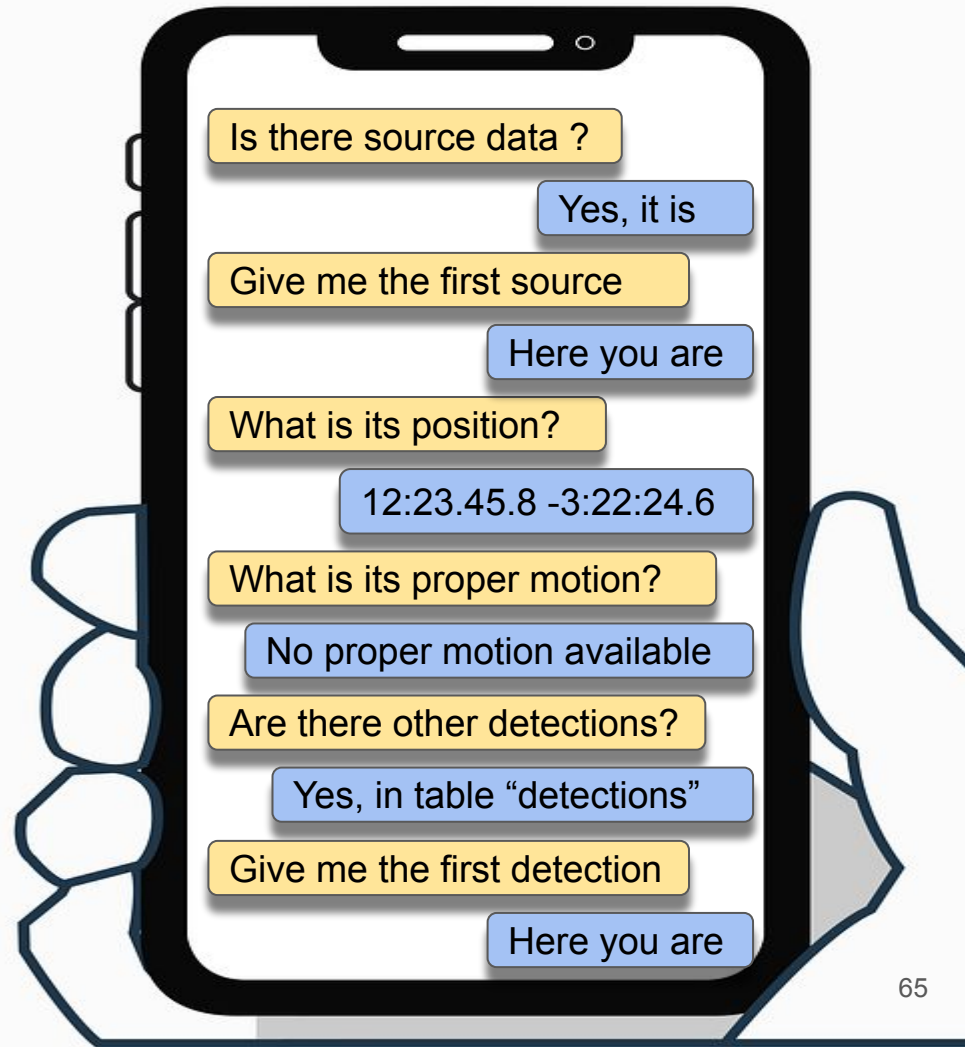
# Client Side Tooling

```
SourceDM — -bash — 80x24
MacBook-Pro-de-Laurent-MICHEL:SourceDM laurentmichel$

parser = Parser("My VoTable")
sources = parser.getInstanceSet("CABMSD");

while( sources.hasNext())
    source = sources.next()
    print(source.get("position"))
    print(source.get("mag.G"))

MacBook-Pro-de-Laurent-MICHEL:SourceDM laurentmichel$
```



# Client Side: Definition of the appropriate API

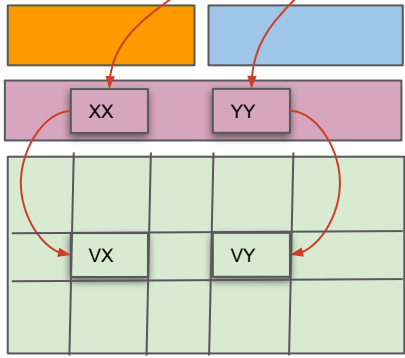
```
<VODML>
  <GLOBALS>
    <...>
    <INSTANCE
      dmid= spaceframe
    </GLOBALS>
  <TEMPLATES>
    <INSTANCE>
      ... dhref= spaceframe
      ... ref=xx ref=yy...
    </INSTANCE>
  </TEMPLATES>
</VODML>
```



```
<INSTANCE>
  <... space frame ...>
  <... val=VX ...?>
  <... val=VY.../>
</INSTANCE>
```



USER



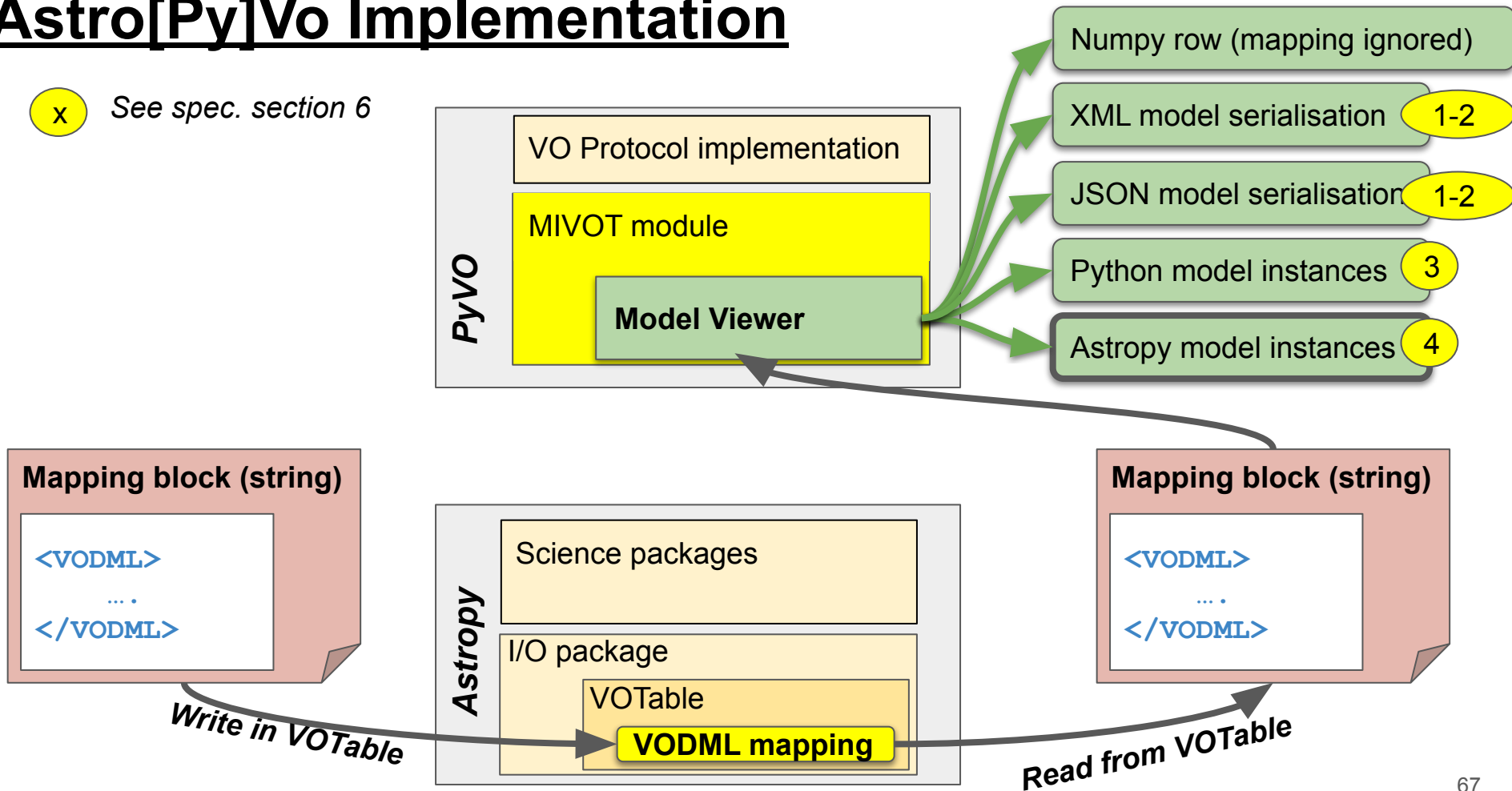
## XML Serialization of the model view of the data rows

- References resolved
- Components gathered

What data structure to provide to the user?

# Astro[Py]Vo Implementation

x See spec. section 6



# Registry

- **Registering MIVOT enabled services**
  - Client need to know whether a service provides annotated data
  - Client need to know which models are mapped by a service provides annotated data
  - Client must be able to ask for data mapped on a particula model
  
- **2 Options considered (Appendix E)**
  - Define new capabilities telling upon which models query results can be mapped
  - Define specific mime types (or TAP format parameter)
    - Used by the XTapDB demonstrator
    - Example: **FORMAT=application/mango**

# Conclusions

- **We have prototypes for every part of the MIVOT workflow**
  - We have shown that this workflow is workable and useful
  - It can also be used to test new models
- **These prototypes must be adapted to be more helpful for the community**
  - Better documentation
  - Better packaging
  - Production quality level
- **We can't do everything on our own**
  - Need active contributors
  - Need testers
  - Need modelers

