

The MIVOT Stack

State of the Art

-

Laurent MICHEL, G.Landais, A. Oberto and G. Mantelet
on the MIVOT/MANGO/Pyvo contributors behalf

What happened in the earlier episodes.

VOTable
1.5

Note that the COOSYS may be deprecated in the future in favor of a more generic way of describing the conventions used to define the positions of the objects studied in the enclosed tables.

Tucson 2023
Closing remarks

Mivot approach for the Epoch propagation L. Michel
Presentation of 4 approach to represent information of COOSYS

VOTable 1.5 Status T. Donaldson
Presentations of evolution and discussion on the process
we go for VOTable 1.5 and let COOSYS to 1.6

https://wiki.ivoa.net/internal/IVOA/InterOpNov2023CloseTCG/apps_cloture.pdf

CSP
Sydney 2024

- Engagement with the community
 - Continuous work on PyVO:
 - Eg. improve the discoverability of astronomical data through the Registry in an end user client such as PyVO
 - Support coordination activities to implement the MIVOT feature into astropy and PyVO Python packages

https://wiki.ivoa.net/internal/IVOA/InterOpMay2024/CSP-Presentation-interop-2024A_FC_2.pdf

Legacy: Position + Proper Motion Weakness

```
<COOSYS ID="J2000" equinox="J2000" epoch="J2000" system="eq_J2000"/>
<FIELD name="pos_RA" ucd="pos.eq.ra;meta.main" datatype="double" unit="deg" ref="J2000"/>
<FIELD name="pos_DEC" ucd="pos.eq.dec;meta.main" datatype="double" unit="deg" ref="J2000"/>
<FIELD name="pm_RA" ucd="pos.pm.ra;meta.main" datatype="double" unit="mas/y" ref="J2000"/>
<FIELD name="pm_DEC" ucd="pos.pm.dec;meta.main" datatype="double" unit="mas/year" ref="J2000"/>
```



- Can see with UCDs that **pos_RA** and **pos_DEC** do work together
- Can see with UCDs that **pm_RA** and **pm_DEC** do work together
- The 4 columns refer to the **coosys** element



- The role of the **@ref->@ID** link is implicit (**@ref** to what?)
- No clear way to see that **pos_RA/pos_DEC** and **pm_RA/pm_DEC** relate to the same quantity
- No clear way to list all **FIELDS** that are connected to **coosys**
- No clear way to associate values with their errors

A Model-based Solution

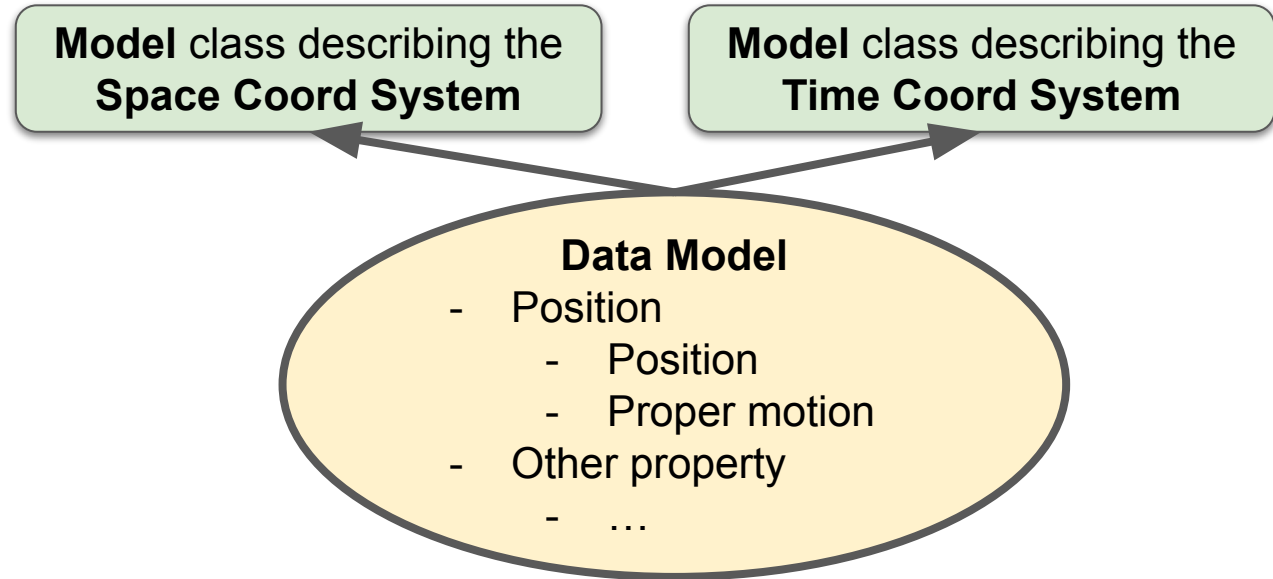
- **VO Standards**
 - **Model into the VOTable**
 - **MIVOT (REC)** to map VOTable **data** to MANGO
 - **Model**
 - **MANGO** (RFC) as a **model** (with Meas/Coords/PhotDM imports)
- **Python Code**
 - `Pyvo.mivot` package (**Pyvo** 1.8+)
 - Read or write annotations
- **Services**
 - **Vizier/XTapDB/FPS/Simbad** to show that **services** can work with this framework

A model Based Solution

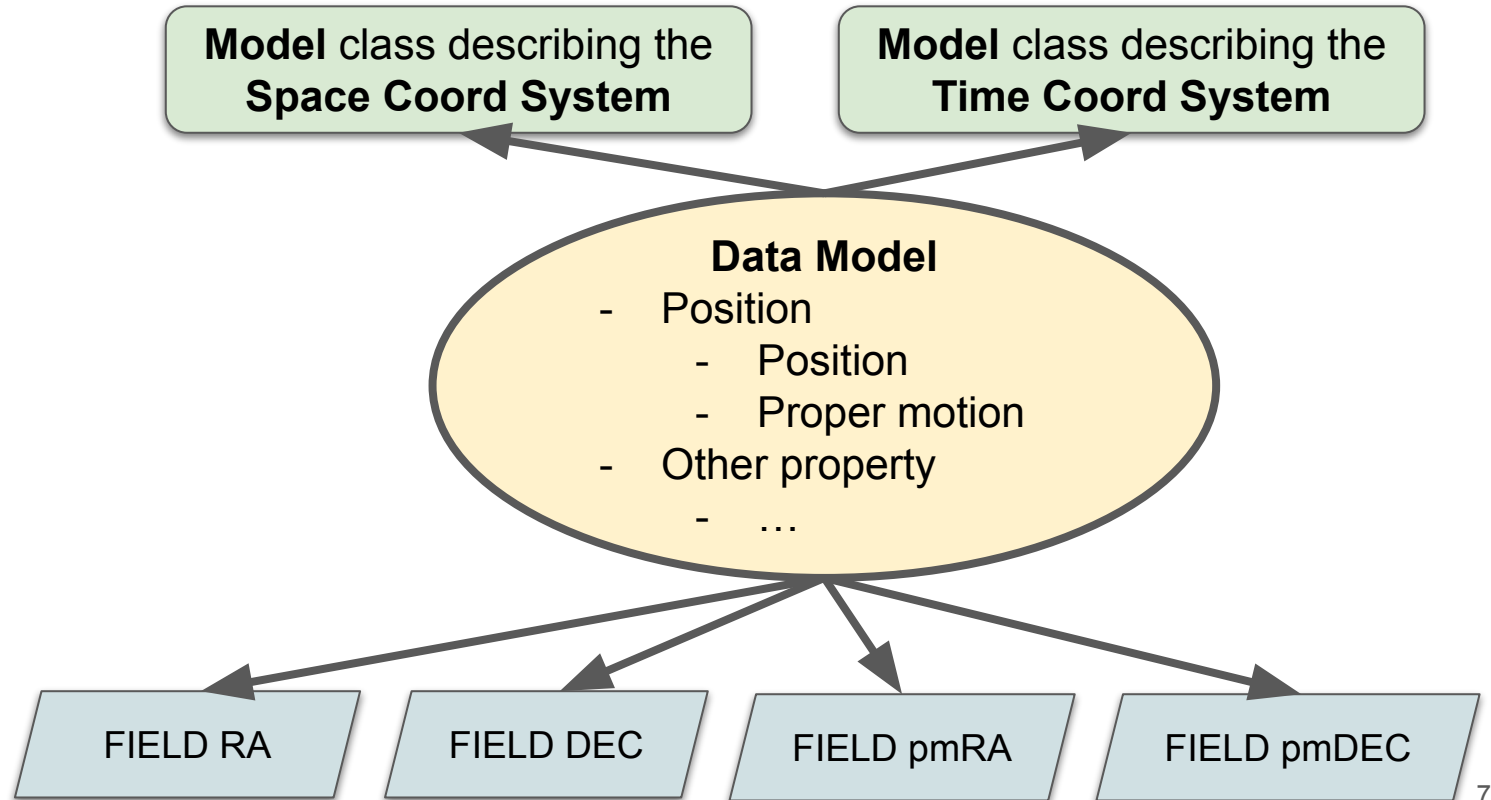
Data Model

- Position
 - Position
 - Proper motion
- Other property
 - ...

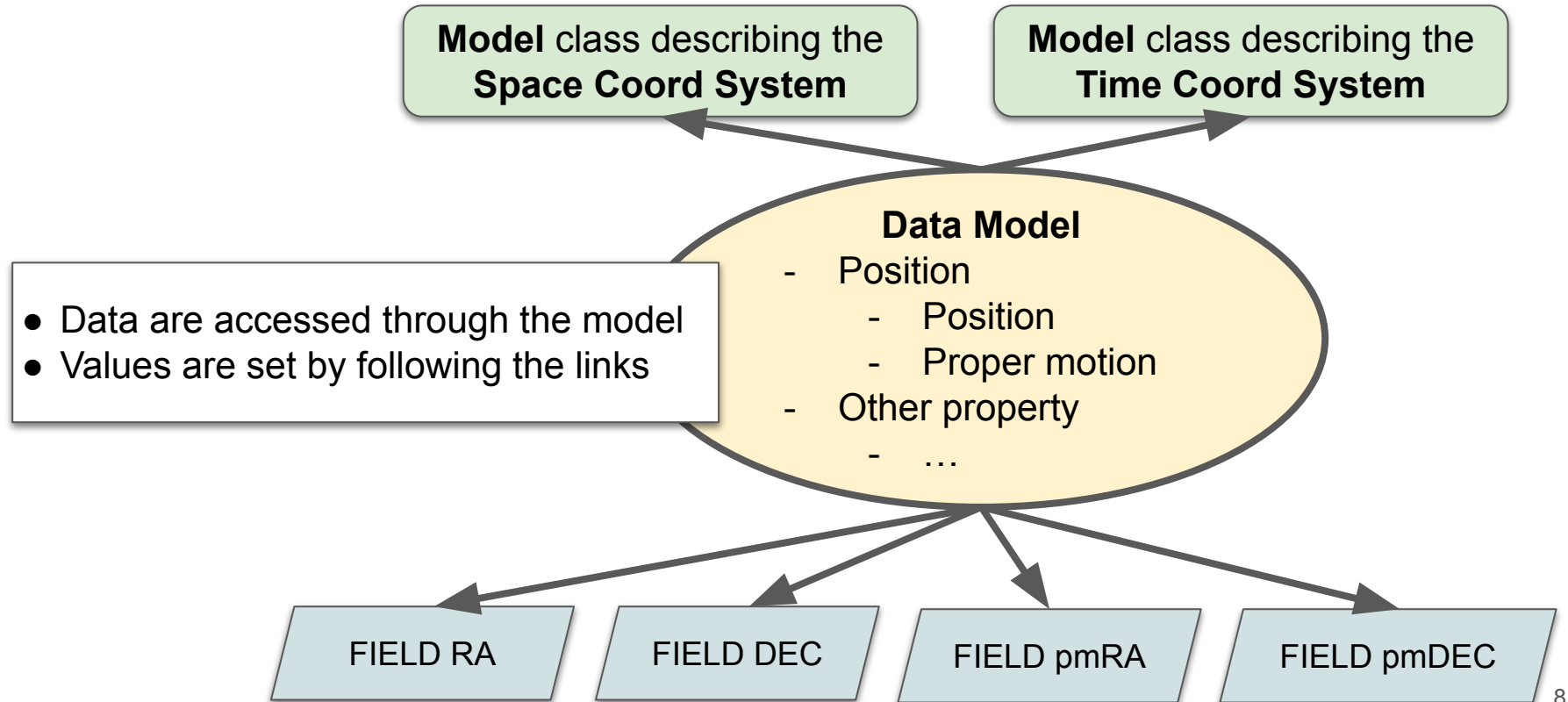
A model Based Solution



A model Based Solution



A model Based Solution



MIVOT: Mapping Block above the Data Table

- Separate RESOURCE before TABLE
- Obey to its one schema

- The space coordinate system is a **GLOBAL** object that can be referenced by any other MIVOT element

- Each table row can be interpreted as an instance of the class **EpochPosition** of the MANGO model

```
<VODML xmlns:dm-mapping="http://www.ivoa.net/xml/mivot">
  <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="mango" />
  <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">
        <INSTANCE dmrole="coords:SpaceFrame.refPosition" dmtype="coords:StdRefLocation">
          <ATTRIBUTE dmrole="coords:StdRefLocation.position" dmtype="ivoa:string" value="NoSet" />
        </INSTANCE>
        <ATTRIBUTE dmrole="coords:SpaceFrame.spaceRefFrame" dmtype="ivoa:string" value="ICRS" />
      </INSTANCE>
    </INSTANCE>
  </GLOBALS>
  <TEMPLATES>
    <INSTANCE dmtype="mango:EpochPosition">
      <REFERENCE dmrole="coords:Coordinate.coosys" dmref="SpaceFrame_ICRS"/>
      <ATTRIBUTE dmrole="mango:EpochPosition.longitude" dmtype="ivoa:RealQuantity" ref="pos_RA"/>
      <ATTRIBUTE dmrole="mango:EpochPosition.latitude" dmtype="ivoa:RealQuantity" ref="pos_DEC" />
      <ATTRIBUTE dmrole="mango:EpochPosition.pmLongitude" dmtype="ivoa:RealQuantity" ref="pm_RA" />
      <ATTRIBUTE dmrole="mango:EpochPosition.pmLatitude" dmtype="ivoa:RealQuantity" ref="pm_DEC"/>
      <ATTRIBUTE dmrole="mango:EpochPosition.pmCosDeltApplied" dmtype="ivoa:boolean" value="true"/>
      <ATTRIBUTE dmrole="mango:EpochPosition.radialVelocity" dmtype="ivoa:RealQuantity" ref="RV"/>
      <ATTRIBUTE dmrole="mango:EpochPosition.parallax" dmtype="ivoa:RealQuantity" ref="PARALLAX" />
      <ATTRIBUTE dmrole="mango:EpochPosition.epoch" dmtype="coords:Epoch" value="J2016.0" unit="year"/>
    </INSTANCE>
  </TEMPLATES>
</VODML>
```

PyVO: Get SkyCoord Instances from a Dataset

```
# run a regular SCS query against SIMBAD
scs_srv = SCSService("https://simbad.cds.unistra.fr/cone")

query_result = scs_srv.search(
    pos=SkyCoord(ra=269.452076 * u.degree,
                 dec=4.6933649 * u.degree, frame='icrs'),
    radius=0.1,
    verbosity=3,
    responseformat="mivot",
    maxrec=10)

# Give the query result to the MIV0t viewer which
# generates the model view of the data
m_viewer = MivotViewer(query_result, resolve_ref=True)

while m_viewer.next_row_view():
    # get the Python object (MangoObject) representing the current row
    mivot_instance = m_viewer.dm_instance
    # Iterate over the properties of the MangoObject
    for mango_property in mivot_instance.propertyDock:
        # build SkyCoord for EpochPosition properties
        if mango_property.dmtypes == "mango:EpochPosition":
            scb = SkyCoordBuilder(mivot_instance)
            print(scb.build_sky_coord())
```

```
<SkyCoord (ICRS): (ra, dec, distance) in (deg, deg, pc)
    (269.45654263, 4.70695153, 6373.486297)
(pm_ra_cosdec, pm_dec, radial_velocity) in (mas / yr, mas / yr, km / s)
    (-4.595, -6.42, 12.434)>

<SkyCoord (ICRS): (ra, dec, distance) in (deg, deg, pc)
    (269.46645811, 4.70561135, 551.26791621)
(pm_ra_cosdec, pm_dec, radial_velocity) in (mas / yr, mas / yr, km / s)
    (0.578, -8.671, 32.35659945)>

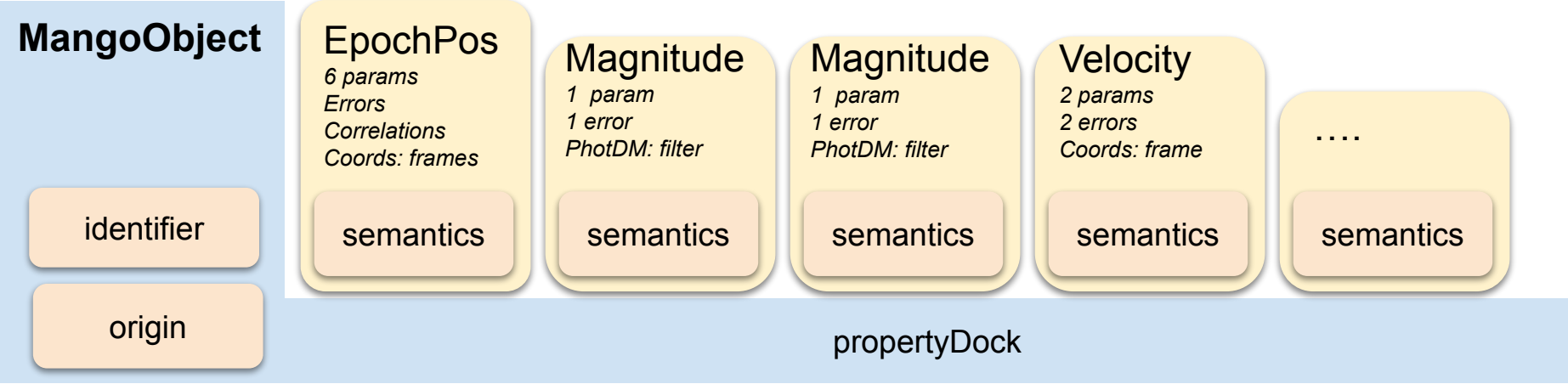
<SkyCoord (ICRS): (ra, dec) in deg
    (269.4562708, 4.6652805)>

<SkyCoord (ICRS): (ra, dec, distance) in (deg, deg, pc)
    (269.47822813, 4.67340155, 444.36544614)
(pm_ra_cosdec, pm_dec) in mas / yr
    (0.56, -8.412)>

<SkyCoord (ICRS): (ra, dec) in deg
    (269.44908, 4.73172)
(radial_velocity) in km / s
    (-110.44,)>
```

Model : MANGO in 2 Slides

- Restore multi-column quantities
- Meta-data enhancement, possibly
- Connect properties to each other



Model : Properties supported by MANGO

Mango Property	Associated frame(s)	error	Role
EpochPosition	coords:Time/SpaceSys	X	Full sky position: 6 parameters + correlations
Brighness	photDM:PhotCal	X	Observed brightness
Color	photDM:PhotFilter	X	Color of the MangoObject.
Status	StatusValues		status defined by a integer number
Label			Free text label
PhysicalProperty	See Measurement DM		Holder for quantities modeled by MeasurementsDM classes
FootPrint	coords:SpaceSys		spatial extent of the MANGO object
DataLink			static links pointing on external data

Read Mivot instances with Dynamic Python Objects

```
m_viewer = MivotViewer(votable, resolve_ref=True)
while m_viewer.next_row_view():
    mivot_instance = m_viewer.dm_instance
    if mivot_instance.dmtypes == "mango:MangoObject":
        print(f"Read source {mivot_instance.identifier.value}")
        for mango_property in mivot_instance.propertyDock:
            if mango_property.dmtypes == "mango:Brightness":
                # get magnitude value and error
                mag_value = mango_property.value.value
                mag_error = mango_property.error.sigma.value
                # get photometric calibration info
                phot_cal = mango_property.photCal
                mag_wl = phot_cal.photometryFilter.spectralLocation.value.value
                mag_filter = phot_cal.identifier.value
                print(f"magnitude at {mag_wl} Angstrom (filter "
                    f"{mag_filter}) is {mag_value:.2f} +/- {mag_error:.2f}")
```

Read source 458-007880

magnitude at 12350.0 Angstrom (filter 2MASS/2MASS.J/AB) is 15.38 +/- 0.04

Let's Have a Look to a Complex TAP Example

```
# run a regular TAP query against XtapDB
URL = "https://xcatdb.unistra.fr/xtapdb"
FORMAT = "application/x-votable+xml;content=mivot"
QUERY = 'SELECT TOP 5 * FROM "public".mergedentry'

service = TAPService(URL)
result = service.run_sync(
    QUERY,
    format=FORMAT)
```

Let's Have a Look at a Complex TAP Example

```
<VODML>
<REPORT status="OK"></REPORT>
<MODEL name="ivoa" url="https://www.ivoa.net/xml/VODML/IVOA-v1.vo-dml.xml"></MODEL>
<MODEL name="coords" url="https://www.ivoa.net/xml/STC/20200908/Coords-v1.0.vo-dml.xml"></MODEL>
<MODEL name="meas" url="https://www.ivoa.net/xml/Meas/20200908/Meas-v1.0.vo-dml.xml"></MODEL>
<MODEL name="phot" url="https://ivoa.net/xml/VODML/Phot-v1.vodml.xml"></MODEL>
<MODEL name="mango" url="https://github.com/ivoa-std/MANGO/raw/refs/heads/wd-v1.0/vo-dml/mango.vo-dml.xml"></MODEL>
<GLOBALS>
<INSTANCE dmid="CoordSystem_ICRS_id" dmrole="" dmtype="coords:SpaceSys"></INSTANCE>
<INSTANCE dmid="CoordSystem_SUM_FLAG_id" dmtype="mango:StatusValues"></INSTANCE>
<INSTANCE dmid="CoordSystem_TCB_id" dmtype="coords:TimeSys"></INSTANCE>
<INSTANCE dmid="CoordSystem_XMM_EB1_id" dmtype="Phot:PhotCal"></INSTANCE>
<INSTANCE dmid="CoordSystem_XMM_EB2_id" dmtype="Phot:PhotCal"></INSTANCE>
<INSTANCE dmid="CoordSystem_XMM_EB3_id" dmtype="Phot:PhotCal"></INSTANCE>
<INSTANCE dmid="CoordSystem_XMM_EB4_id" dmtype="Phot:PhotCal"></INSTANCE>
<INSTANCE dmid="CoordSystem_XMM_EB5_id" dmtype="Phot:PhotCal"></INSTANCE>
<INSTANCE dmid="CoordSystem_XMM_EB8_id" dmtype="Phot:PhotCal"></INSTANCE>
<INSTANCE dmid="CoordSystem_XMM_FILTER_EB1_id" dmtype="Phot:PhotometryFilter"></INSTANCE>
<INSTANCE dmid="CoordSystem_XMM_FILTER_EB2_id" dmtype="Phot:PhotometryFilter"></INSTANCE>
<INSTANCE dmid="CoordSystem_XMM_FILTER_EB3_id" dmtype="Phot:PhotometryFilter"></INSTANCE>
<INSTANCE dmid="CoordSystem_XMM_FILTER_EB4_id" dmtype="Phot:PhotometryFilter"></INSTANCE>
<INSTANCE dmid="CoordSystem_XMM_FILTER_EB5_id" dmtype="Phot:PhotometryFilter"></INSTANCE>
<INSTANCE dmid="CoordSystem_XMM_FILTER_EB8_id" dmtype="Phot:PhotometryFilter"></INSTANCE>
</GLOBALS>
<TEMPLATES tableref="Results">
<INSTANCE dmtype="mango:MangoObject">
<ATTRIBUTE dmrole="mango:MangoObject.identifier" dmtype="ivoa:string" ref="IAUNAME"></ATTRIBUTE>
<COLLECTION dmrole="mango:MangoObject.propertyDock">
<INSTANCE dmrole="" dmtype="mango:Status"></INSTANCE>
<INSTANCE dmrole="" dmtype="mango:EpochPosition"></INSTANCE>
<INSTANCE dmrole="" dmtype="mango:Color"></INSTANCE>
<INSTANCE dmrole="" dmtype="mango:Color"></INSTANCE>
<INSTANCE dmrole="" dmtype="mango:Color"></INSTANCE>
<INSTANCE dmrole="" dmtype="mango:Color"></INSTANCE>
<INSTANCE dmrole="" dmtype="mango:Color"></INSTANCE>
<INSTANCE dmrole="" dmtype="mango:Brightness"></INSTANCE>
<INSTANCE dmrole="" dmtype="mango:Brightness"></INSTANCE>
<INSTANCE dmrole="" dmtype="mango:Brightness"></INSTANCE>
<INSTANCE dmrole="" dmtype="mango:Brightness"></INSTANCE>
<INSTANCE dmrole="" dmtype="mango:Brightness"></INSTANCE>
<INSTANCE dmrole="" dmtype="mango:Brightness"></INSTANCE>
</COLLECTION>
</INSTANCE>
</TEMPLATES>
</VODML>
```

```
against XtapDB
stra.fr/xtapdb"
votable+xml;content=mivot"
FROM "public".mergedentry'
```

Let's Have a Look at a Complex TAP Example

<GLOBALS>

<INSTANCE dmid="CoordSystem_ICRS_id" dmrole="" dmtype="coords:SpaceSys"></INSTANCE>

<INSTANCE dmid="CoordSystem_SUM_FLAG_id" dmtype="mango:StatusValues"></INSTANCE>

<INSTANCE dmid="CoordSystem_TCB_id" dmtype="coords:TimeSys"></INSTANCE>

<INSTANCE dmid="CoordSystem_XMM_EB1_id" dmtype="Phot:PhotCal"></INSTANCE>

<INSTANCE dmid="CoordSystem_XMM_EB2_id" dmtype="Phot:PhotCal"></INSTANCE>

<INSTANCE dmid="CoordSystem_XMM_EB3_id" dmtype="Phot:PhotCal"></INSTANCE>

<INSTANCE dmid="CoordSystem_XMM_EB4_id" dmtype="Phot:PhotCal"></INSTANCE>

<INSTANCE dmid="CoordSystem_XMM_EB5_id" dmtype="Phot:PhotCal"></INSTANCE>

<INSTANCE dmid="CoordSystem_XMM_EB8_id" dmtype="Phot:PhotCal"></INSTANCE>

<INSTANCE dmid="CoordSystem_XMM_FILTER_EB1_id" dmtype="Phot:PhotometryFilter"></INSTANCE>

<INSTANCE dmid="CoordSystem_XMM_FILTER_EB2_id" dmtype="Phot:PhotometryFilter"></INSTANCE>

<INSTANCE dmid="CoordSystem_XMM_FILTER_EB3_id" dmtype="Phot:PhotometryFilter"></INSTANCE>

<INSTANCE dmid="CoordSystem_XMM_FILTER_EB4_id" dmtype="Phot:PhotometryFilter"></INSTANCE>

<INSTANCE dmid="CoordSystem_XMM_FILTER_EB5_id" dmtype="Phot:PhotometryFilter"></INSTANCE>

<INSTANCE dmid="CoordSystem_XMM_FILTER_EB8_id" dmtype="Phot:PhotometryFilter"></INSTANCE>

</GLOBALS>

Let's Have a Look at a Complex TAP Example

```
# run a regular TAP query against XtapDB
URL = "https://xcatdb.unistra.fr/xtapdb"
FORMAT = "application/x-votable+xml;content=mivot"
QUERY = 'SELECT TOP 5 * FROM "public".mergedentry'
```

```
</GLOBAL>
<TEMPLATES tableref="Results">
  <INSTANCE dmtype="mango:MangoObject">
    <ATTRIBUTE dmrole="mango:MangoObject.identifier" dmtype="ivoa:string" ref="I
    <COLLECTION dmrole="mango:MangoObject.propertyDock">
      <INSTANCE dmrole="" dmtype="mango:Status"><input type="checkbox"/></INSTANCE>
      <INSTANCE dmrole="" dmtype="mango:EpochPosition"><input type="checkbox"/></INSTANCE>
      <INSTANCE dmrole="" dmtype="mango:Color"><input type="checkbox"/></INSTANCE>
      <INSTANCE dmrole="" dmtype="mango:Color"><input type="checkbox"/></INSTANCE>
      <INSTANCE dmrole="" dmtype="mango:Color"><input type="checkbox"/></INSTANCE>
      <INSTANCE dmrole="" dmtype="mango:Color"><input type="checkbox"/></INSTANCE>
      <INSTANCE dmrole="" dmtype="mango:Color"><input type="checkbox"/></INSTANCE>
      <INSTANCE dmrole="" dmtype="mango:Brightness"><input type="checkbox"/></INSTANCE>
      <INSTANCE dmrole="" dmtype="mango:Brightness"><input type="checkbox"/></INSTANCE>
      <INSTANCE dmrole="" dmtype="mango:Brightness"><input type="checkbox"/></INSTANCE>
      <INSTANCE dmrole="" dmtype="mango:Brightness"><input type="checkbox"/></INSTANCE>
      <INSTANCE dmrole="" dmtype="mango:Brightness"><input type="checkbox"/></INSTANCE>
      <INSTANCE dmrole="" dmtype="mango:Brightness"><input type="checkbox"/></INSTANCE>
    </COLLECTION>
  </INSTANCE>
</TEMPLATES>
```

```

# Give the query result to the MIVOT viewer which
# generates the model view of the data
m_viewer = MivotViewer(result, resolve_ref=True)

while m_viewer.next_row_view():
    # get the Python object (MangoObject) representing the current row
    mivot_instance = m_viewer.dm_instance
    sp_location = []
    sp_filter = []
    mag = []
    mag_error = []
    if mivot_instance.dctype == "mango:MangoObject":
        for mango_property in mivot_instance.propertyDock:
            if mango_property.dctype == "mango:Brightness":
                if mango_property.value.value:
                    # get the flux value, error and unit
                    unit = mango_property.value.unit
                    mag.append(mango_property.value.value)
                    mag_error.append(mango_property.error.sigma.value)
                    phot_cal = mango_property.photCal
                    # get the filter spectral location (filter name + location)
                    spectral_location = phot_cal.photometryFilter.spectralLocation
                    sp_location.append(
                        spectral_location.value.value)
                    sp_filter.append(phot_cal.identifier.value)
                    sp_unit = spectral_location.unitexpression.value

    plot_sed(mivot_instance.identifier.value, unit, sp_unit, mag, mag_error,
             sp_filter, sp_location)

```

```
# Give the query result to the MIVOT viewer which  
# generates the model view of the data
```

```
m_viewer = MivotViewer
```

```
while m_viewer.next_r
```

```
    # get the Python
```

```
    mivot_instance =
```

```
    sp_location = []
```

```
    sp_filter = []
```

```
    mag = []
```

```
    mag_error = []
```

```
    if mivot_instance
```

```
        for mango_pro
```

```
            if mango
```

```
                if ma
```

```
                    #
```

```
                    u
```

```
                    m
```

```
                    m
```

```
                    p
```

```
                    #
```

```
                    s
```

```
                    s
```

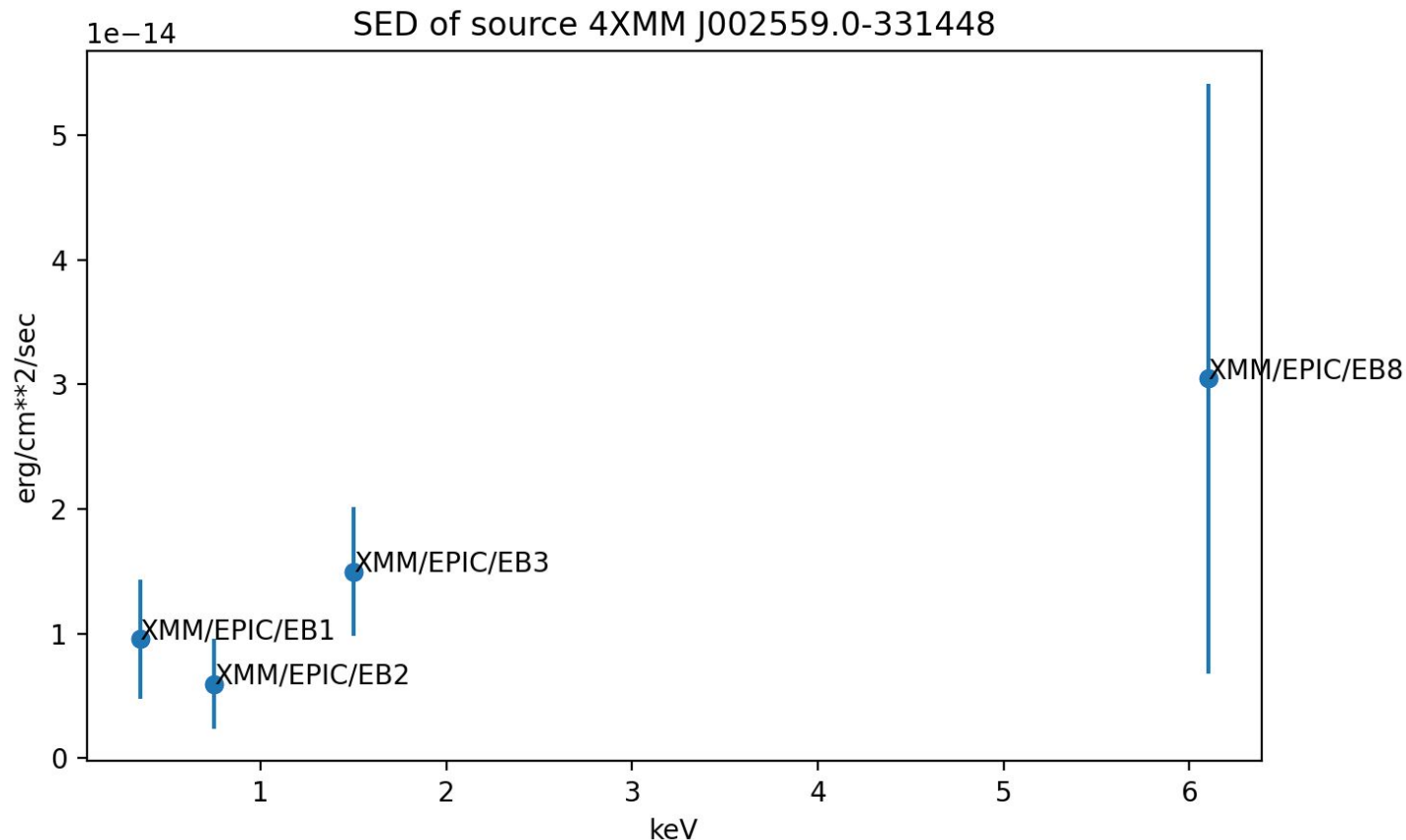
```
                    s
```

```
                    s
```

```
                    s
```

```
                    s
```

```
plot_sed(mivot_instance.identifier.value, unit, sp_unit, mag, mag_error,  
         sp_filter, sp_location)
```



PyVO: Build MIVOT Annotations (1.7+)

```
votable = parse(votable_path)
builder = InstancesFromModels(votable, dmid="URAT1")
```

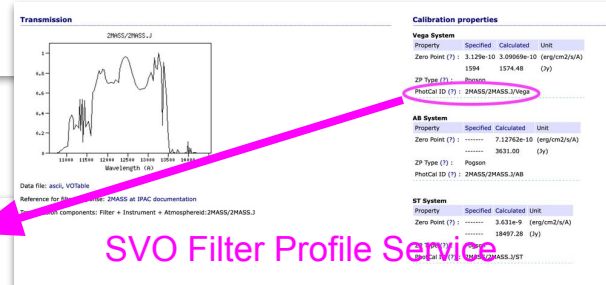
```
⋮
⋮
⋮
# map VOTable columns on individual properties
# Properties are mapped one by one
⋮
```

```
builder.pack_into_votable()
votable.to_xml(output_path)
```

- Python API for building annotations on a parsed VOTable
- Can be help for a service development

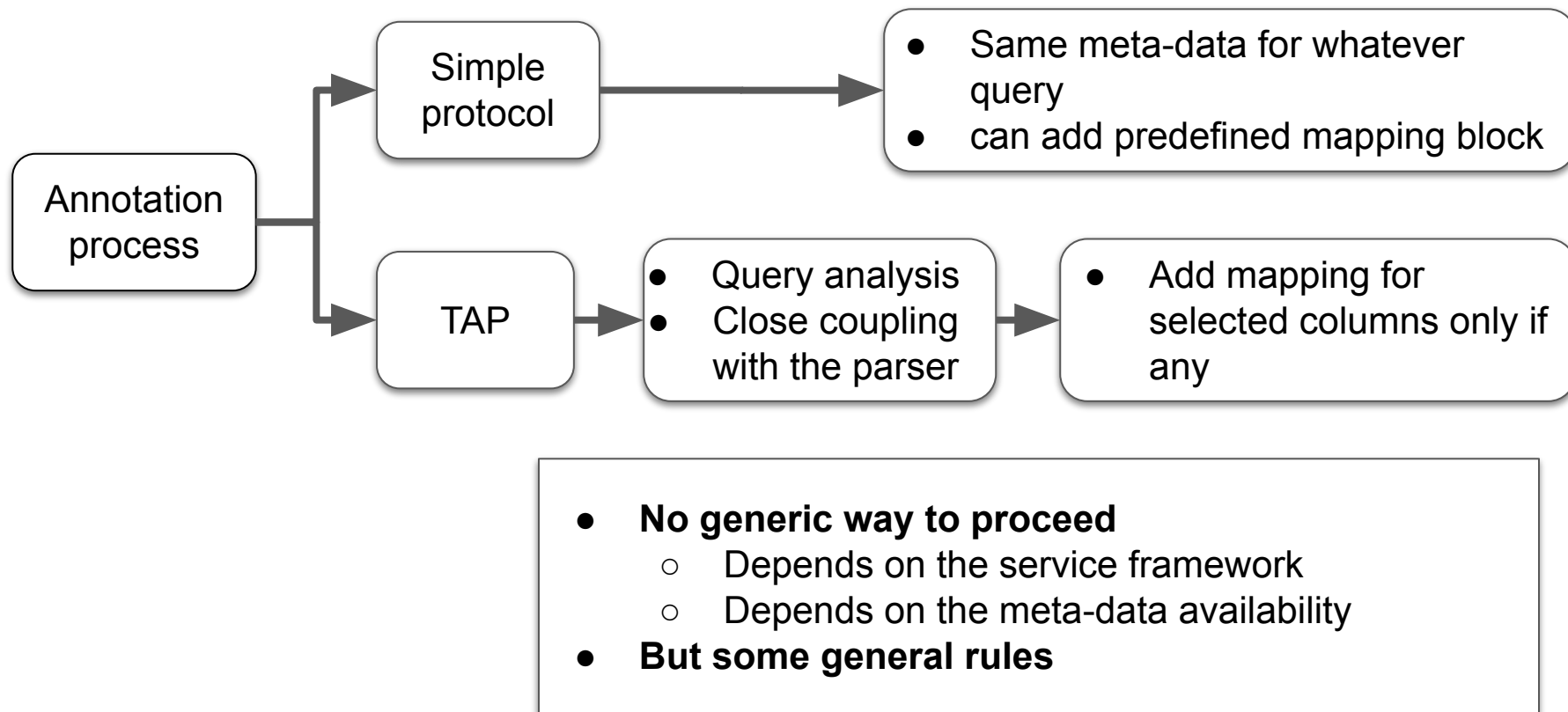
PyVO: map a magnitude (value + error + filter)

```
<FIELD name="Jmag" ucd="phot.mag;em.IR.J" datatype="float" width="6" precision="3" unit="mag"><!-- ucd="PHOT_JHN_J" -->
  <DESCRIPTION>?(Jmag) 2MASS J-band magnitude</DESCRIPTION>
  <VALUES null="NaN" />
</FIELD>
<FIELD name="e_Jmag" ucd="stat.error;phot.mag" datatype="float" width="6" precision="3" unit="mag"><!-- ucd="ERROR" -->
  <DESCRIPTION>?(eimag) Error on Jmag</DESCRIPTION>
  <VALUES null="NaN" />
</FIELD>
```

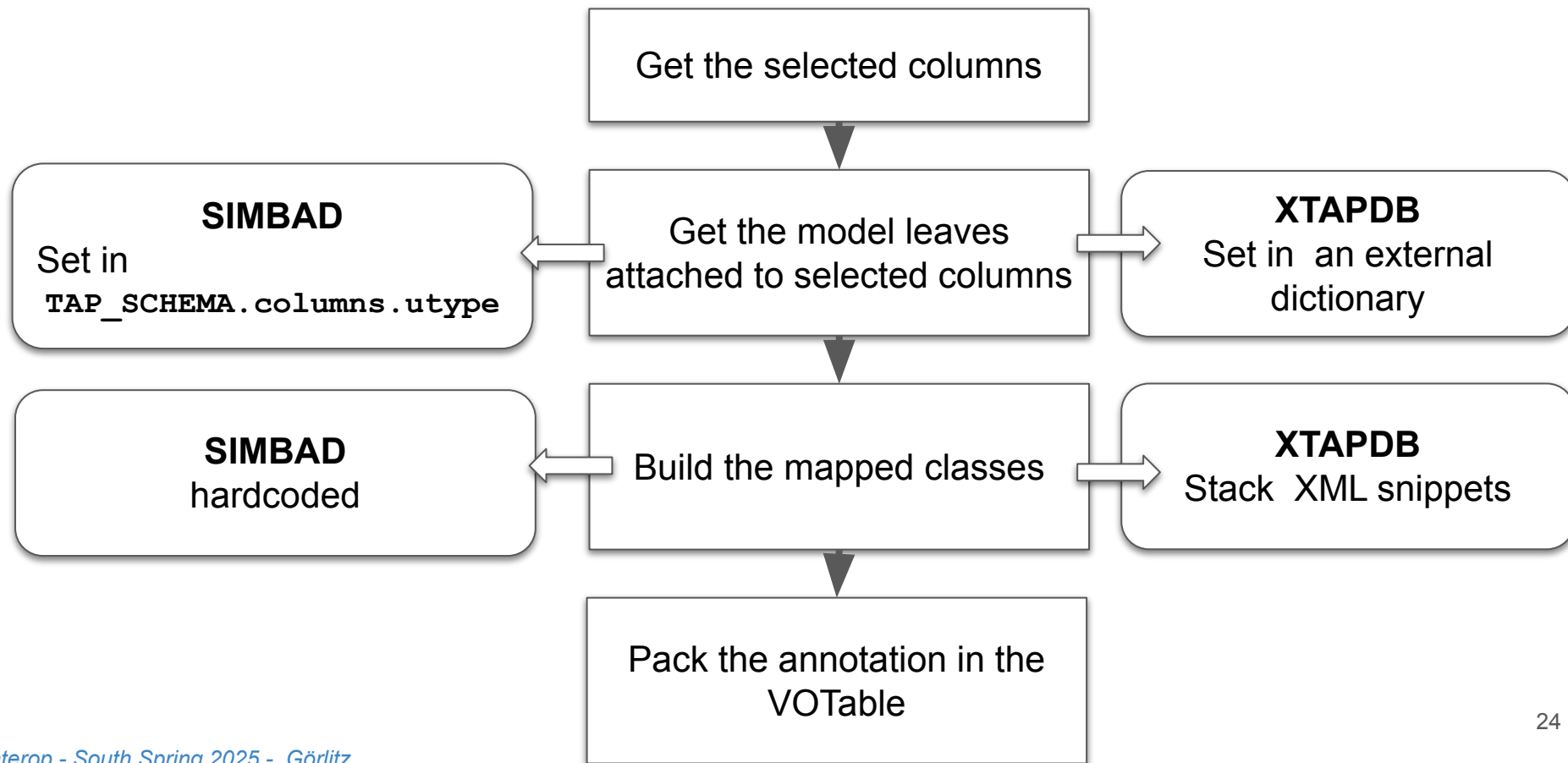


```
builder.add_mango_brightness( photcal_id="2MASS/2MASS.J/AB",
                               mapping={"value": "Jmag",
                                         "error": {"class": "PErrorSym1D", "sigma": "e_Jmag"}},
                               semantics={"description": "magnitude J",
                                         "uri": "https://www.ivoa.net/rdf/uat/2024-06-25/uat.html#magnitude",
                                         "label": "magnitude"})
```

Annotation Process on Server Side



Annotation of TAP Responses (VOLLT)



Interop - South Spring 2025 - Görlitz



Search projects

HelpSponsorsLog inRegister

mivot-validator 0.2

`pip install mivot-validator`

Latest versionReleased: Oct 3, 2023

Tools for validating VOTable annotated with MIVOT and to generate MIVOT snippets

Navigation

Project description

Release history

Download files

Verified details

These details have been *verified by PyPI*



laurent.michel

Unverified details

These details have *not* been verified by PyPI

Project links

[Homepage](#)

Meta

- License: MIT License (MIT License Copyright (c) 2022 International

Project description

mivot-validator

This package has 2 purposes:

- Validation of VOTables annotated with IVOA recommendation [MIVOT](#)
- MIVOT serialization of model components (snippets) that can be used to build annotations

Installation

The validator is distributed as a Python package.

```
$ pip install mivot-validator
```

Scripts for Validating Annotated VOTables

There are 2 validation levels:

- against the XML schemas (VOTable and MIVOT)
- against the model itself as it is defined in VODML

Validator For annotated VOTables

2 Stages validation

- Against VOTable and MIVOT XML schemas
- Against the model structure

Class snippet generator

- XML Snippets for all classes of a model
- XML snippet for one class with constraints on the concrete classes to be instanced

Available on Pipy

- `% pip install mivot-validator`

Web Application

- Feature by feature documentation

MIVOT-Syntax Use Cases and Examples

ATTRIBUTE Mapping Example in MIVOT

The ATTRIBUTE element defines either a class attribute or a collection item, both set with atomic va...

[Go to page](#)[DALI xHtml](#)

COLLECTION Mapping Example in MIVOT

COLLECTION is a container element to be used each time we have to gather items of the same kind. It ...

[Go to page](#)[DALI xHtml](#)

PRIMARY_KEY Mapping Example in MIVOT

The PRIMARY_KEY element allows to set an identification key to an INSTANCE. The primary keys are on...

[Go to page](#)[DALI xHtml](#)

INSTANCE Mapping Example in MIVOT

This is a mapping example in MIVOT for the INSTANCE field. Its purpose is to defines a complex Objec...

[Go to page](#)[DALI xHtml](#)

MODEL Mapping Example in MIVOT

A VOTable can provide serializations for an arbitrary number of data models. In order to declare whi...

[Go to page](#)[DALI xHtml](#)

JOIN Mapping Example in MIVOT

The JOIN element populates a COLLECTION with INSTANCE elements from another collection, called herea...

[Go to page](#)[DALI xHtml](#)

MIVOT-Syntax: ATTRIBUTE Mapping Example in MIVOT

The ATTRIBUTE element defines either a class attribute or a collection item, both set with atomic values. The requirements on the content (especially @ref and @drole, may differ depending on the usage.

Usage

1. Child of INSTANCE
The ATTRIBUTE can be seen as a class attribute; it must have a @drole XML attribute.
In this case, the ATTRIBUTE must be specified by:
 - * @ref - reference to a VOTable PARAM or FIELD
 - * @value - a literal
 - * if both are provided; @value serves as the default if the reference cannot be resolved
 - * if @ref cannot be resolved and there is no @value, the client must consider that value as NULL in its software context (Language, inner model).
2. Child of COLLECTION
In this case the host COLLECTION can be seen as a vector and the ATTRIBUTE as one coordinate of the vector.
It must have no @drole XML attribute or an empty one.
In this case, the ATTRIBUTE must be specified by:
 - * @ref - reference to a VOTable PARAM or FIELD
 - * @value - a literal
 - * if both are provided; @value serves as the default if the reference cannot be resolved
 - * if @ref cannot be resolved and there is no @value, the client must consider that value as NULL.The definition of the NULL value depends of the software environment
3. Any case :
The ATTRIBUTE must always have a non-empty @dtype XML attribute.
This @dtype may complete missing types in the VOTable.
In case of inconsistency with VOTable @dtype SHOULD supersede VOTable types.

<https://saada.unistra.fr/voexamples/show/MIVOT-Syntax>

[Go to page](#)[DALI xHtml](#)[Go to page](#)[DALI xHtml](#)

will be set with the value of the Idobstime column:

Implementations: PyVO

package	role	status
(*) <code>pyvo.mivot.viewer</code>	Read annotation build dynamic objects	Works since 1.6
(*) <code>pyvo.mivot.feature</code>	SkyCoord builder Mango interface	Works since 1.8 (**) Still miss a MANGO interface
(*) <code>pyvo.mivot.writer</code>	API for building annotations	Works since 1.7 (**) Still miss a MANGO interface

(*) Still as a prototype feature until MANGO is a REC
(**) Needs MANGO To be a REC to be completed

Implementations : Services

Property	Vizier SCS	Slmbad SCS	XTapDB	Simbad TAP (*)
EpochPosition	x	x	x	x
Brightness			x	x (**)
Color			x	
Status		x	x	x

(*) To be deployed once MANGO in REC

(**) Problem with the photometric calibration mapping due to the fact that the photometric system may change from an object to another.

STATUS AND PROSPECTS

All the implementations shown here demonstrate that

- the model approach works well
- it can be used.

Waiting MANGO is a REC before to extend the PyVO implementation

- MANGO-centric feature
- More robust

Available in Pyvo

- Viewer: 1.6+
- Writer: 1.7+
- Enhanced viewer: 1.8

Hope to have new implementations

- services
- client codes

STATUS AND PROSPECTS

All the implementations shown here demonstrate that

- the model approach works
- And can be used.

Waiting MANGO is a RESEARCH

- MANGO

*Many thanks to all of the contributors,
especially the PyVO maintainers the
Simbad and the Vizier team*

in Pyvo

- viewer: 1.6+
- viewer: 1.7+
- Enhanced viewer: 1.8

Hope to have new implementations

- services
- client codes

Resources

Vizier SCS :

<https://vizier.cds.unistra.fr/viz-bin/conesearch/V1.5/>

Simbad SCS :

<https://simbad.cds.unistra.fr/cone?verb=2&responseformat=mivot>

XTapDB (see full URL on the RFC page)

<https://xcatdb.unistra.fr/xtapdb> (FORMAT = "application/x-votable+xml;content=mivot")

ONLINE DOCUMENTATION

<https://saada.unistra.fr/voexamples/show/MIVOT-Syntax>

Notebook(s) (work in progress)

https://github.com/lmichel/pyvo_mivot_bench

MANGO RFC page

<https://wiki.ivoa.net/twiki/bin/view/IVOA/Mango10RFC>

History Page



- 11/2025 Improvement of the **PyVO** interpreter (1.8)
- 09/2025 **Simbad** deploys a cone search returning annotated data
- 07/2025 **MANGO** in RFC
- 06/2025 **PyVO** (1.7) API for building annotations
- 06/2025 **TAP service** returning annotated **XMM data**
- 03/2024 **PyVO** (1.6) can read VOTable data through the MIVOT annotation
- 02/2024 **Vizier** deploys a cone search returning annotated data
- 02/2024 **Astropy** 6 enable to read and write **MIVOT** annotations
- 06/2023 **MIVOT** IVOA Recommendation
- 2022 **MIVOT** serialization of **PhotDM** by the filter profile service
- 2022 BoF **ADASS** : TAP and the data-models
- 2021 **DM-WG** workshop

MODEL

L. Michel, M. Cresitello Dittmar, J.Salgado, G. Landais, M. Molinaro, M. Louys, F. Bonnarel

SERVER SIDE

L. Michel, G. Landais, G. Mantelet, A. Oberto, I. Errami, Carlos Rodriguo (rip)

CLIENT SIDE

L. Michel, F.X. Pineau, J.Abid, S. Floret, M. Marchand, M. Taylor, Astropy/Pyvo team

VALIDATOR

L. Michel, J.Abid, M. Louys, F. Bonnarel,

ONLINE DOCUMENTATION

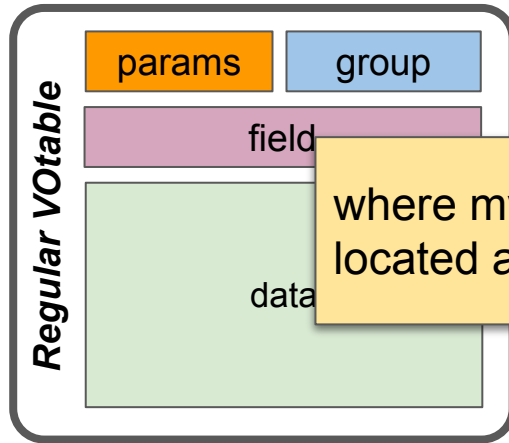
L. Michel, J.Abid, M. Louys, F. Bonnarel, D. Durand

MISCELLANEOUS

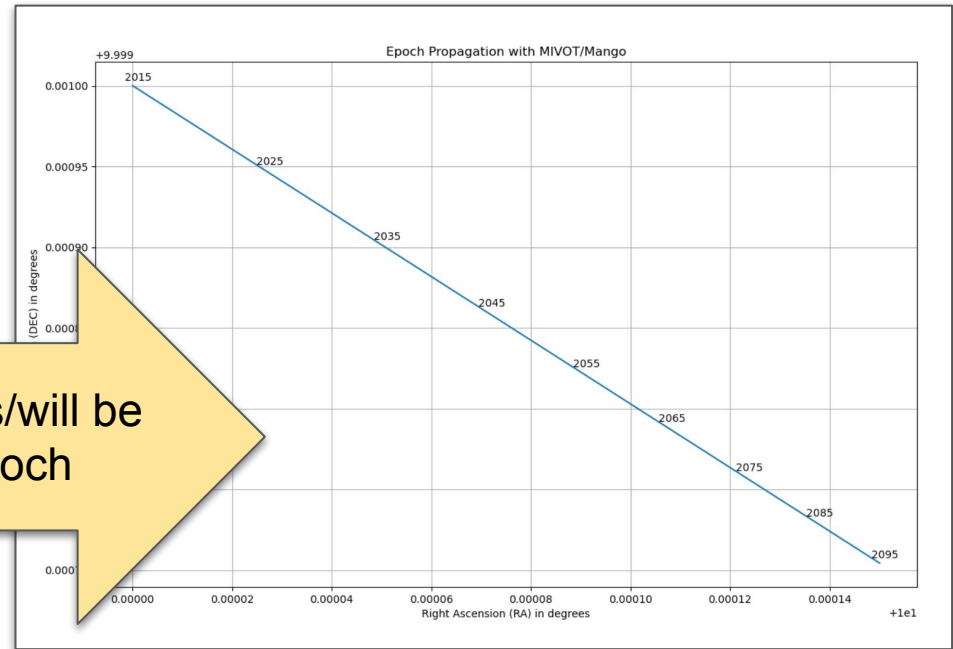
L. Michel, T. Boch, P. Fernique, M. Louys, F. Bonnarel, C. Nogueira

backup

The Challenge



where my object was/will be located at a given epoch



In this VOTABLE, I have somewhere:

- Position
- Proper motion
- Parallax
- Radial velocity
- Desired metadata

The challenge: Agree on an appropriate way to present data to make the process standard

Legacy: Position + Proper Motion

```
<COOSYS ID="J2000" equinox="J2000" epoch="J2000" system="eq_J2000"/>
<FIELD name="pos_RA" ucd="pos.eq.ra;meta.main" datatype="double" unit="deg" ref="J2000"/>
<FIELD name="pos_DEC" ucd="pos.eq.dec;meta.main" datatype="double" unit="deg" ref="J2000"/>
<FIELD name="pm_RA" ucd="pos.pm.ra;meta.main" datatype="double" unit="mas/y" ref="J2000"/>
<FIELD name="pm_DEC" ucd="pos.pm.dec;meta.main" datatype="double" unit="mas/year" ref="J2000"/>
```



- Can see with UCDs that **pos_RA** and **pos_DEC** do work together
- Can see with UCDs that **pm_RA** and **pm_DEC** do work together
- The 4 columns refer to the **COOSYS** element



- The role of the **@ref->@ID** link is implicit (**@ref** to what?)
- No clear way to see that **pos_RA/pos_DEC** and **pm_RA/pm_DEC** relate to the same quantity

4 RUST implementation (F.X Pineau J. Abid)

```
use std::path::Path;
use crate::mivot::ModelLayer;

let mut model_layer = ModelLayer::from_file(Path::new("my-votable.xml"), true).unwrap();

model_layer.init_epoch_positions();

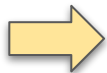
// Retrieve the EpochPosition instance from the mapping block
let epoch_positions = model_layer.get_epoch_positions_as_ref().get("EpochPosition").unwrap();

// Browse the instance
print!("{}", epoch_positions.longitude.value);
print!("{}", epoch_positions.latitude.value);
print!("{}", epoch_positions.coosys.frame.spaceRefFrame);
```

This implementation is model-dependant
New/change model => New code

MIVOT: Flexibility

EPOCH defined in a **<FIELD>**
@ref to the FIELD identifier



```
<ATTRIBUTE dmrole="tucson:Position.epoch"  
           dmtype="coords:Epoch"  
           unit="year"  
           ref="_EPOCH_FIELD"/>
```

EPOCH defined in a **<PARAM>**
@ref to the PARAM identifier



```
<ATTRIBUTE dmrole="tucson:Position.epoch"  
           dmtype="coords:Epoch"  
           unit="year"  
           ref="_EPOCH_PARAM"/>
```

EPOCH defined in a non
machine-readable element
No @ref but a fixed @value



```
<ATTRIBUTE dmrole="tucson:Position.epoch"  
           dmtype="coords:Epoch"  
           unit="year"  
           value="J2023.88"/>
```

The structure of MIVOT block is not altered by the way the EPOCH is set in the VOTable

- Allow server code to be versatile
- Allow a same client code to process many different VOTable