

The MIVOT Stack

PyVO Focus

-
Laurent MICHEL 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

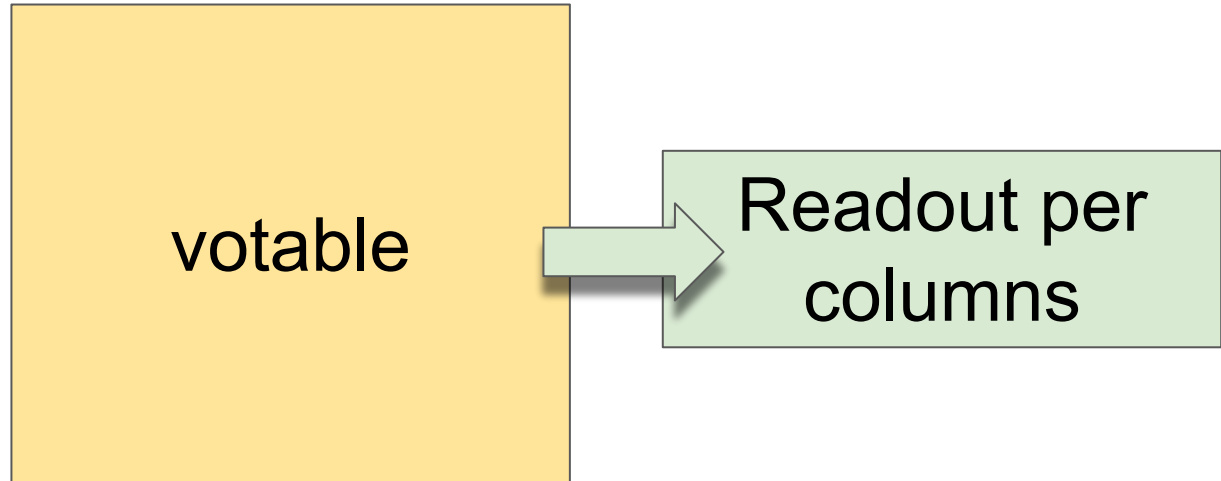
- Engagement with the community
 - Continuous work on PyVO:
 - Eg. improve the discoverability of astronomical data through the Registry in an end user client such as PyVO
 - Support coordination activities to implement the MIVOT feature into astropy and PyVO Python packages

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

A Model-based Solution

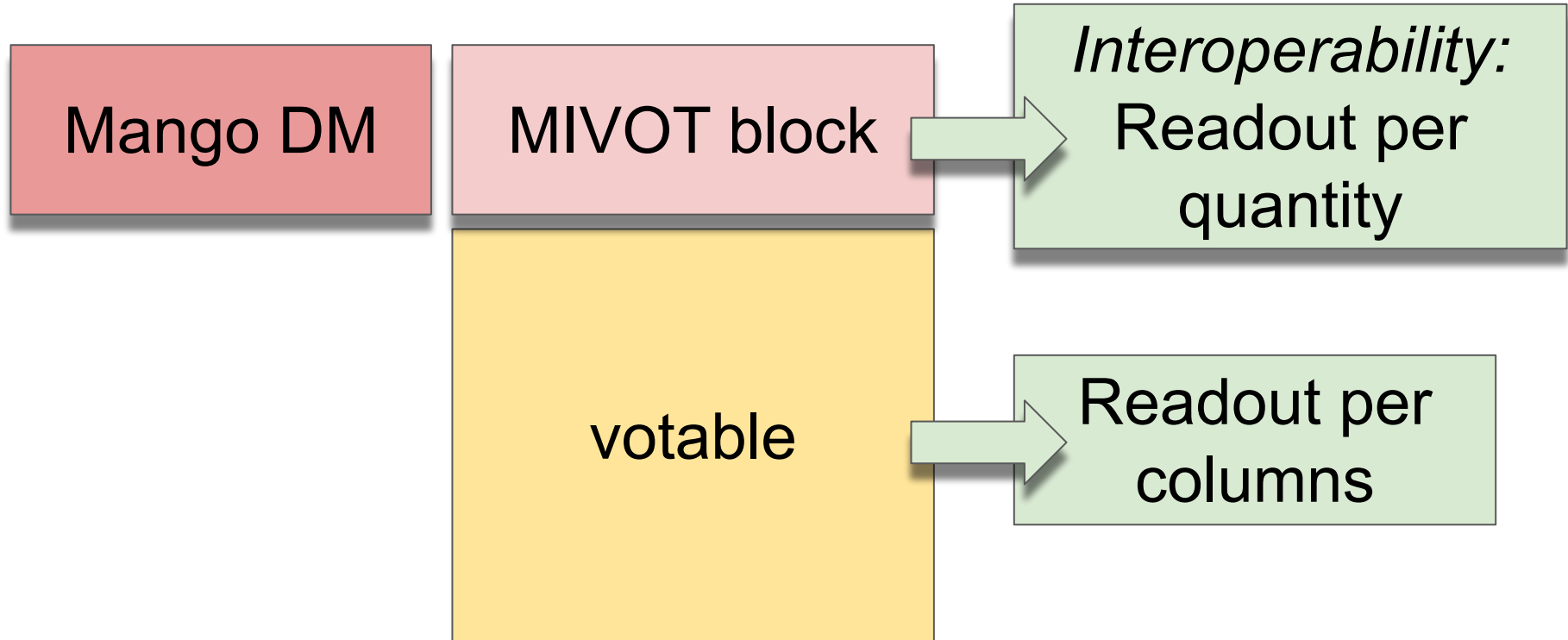
- **VO Standards**
 - **MANGO** as a **model** (with Meas/Coords/PhotDM imports)
 - **MIVOT** to **map** VOTable **data** to MANGO
- **Tools**
 - **PYVO** to read/write **MIVOT** annotations (see `pyvo.mivot`)
 - **MIVOT validator** to make sure the job is well done
- **Services**
 - **Vizier/XCatDB/FPS** to show that **services** can work with this framework

The Workflow in One Slide



The Workflow in One Slide

vv



The Workflow in One Slide

Presented in
DM session

Mango DM

How to build it?

MIVOT block

Which code can
do that?

Interoperability:
Readout per
quantity

votable

Readout per
columns

MANGO in one Slide

MangoObject

identifier

origin

position

semantics

magnitude

semantics

magnitude

semantics

velocity

semantics

....

semantics

propertyDock

Restore multi-column quantities

Meta-data enhancement, possibly

Connect properties to each other

MIVOT: Mapping Block above the Data Table

- 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>
```

Get SkyCoord Object through MIVOT/MANGO

```
# Enable MIVOT-specific features in the pyvo library
activate_features("MIVOT")

scs_srv = SCSService("https://vizier.cds.unistra.fr/viz-bin/conesearch/V1.5/I
query_result = scs_srv.search(
    pos=SkyCoord(ra=52.26708 * u.degree, dec=59.94027 * u.degree, frame='icrs
    radius=0.5)

# The MIVOT viewer generates the model view of the data
m_viewer = MivotViewer(query_result, resolve_ref=True)
```

I/239/hip_main: hipparcos

Get SkyCoord Object through MIVOT/MANGO

```
# Enable MIVOT-specific features in the pyvo library
activate_features("MIVOT")

scs_srv = SCSService("https://vizier.cds.unistra.fr/viz-bin/conesearch/V1.5/I

query_result = scs_srv.search(
    pos=SkyCoord(ra=52.26708 * u.degree, dec=59.94027 * u.degree, frame='icrs',
    radius=0.5)

# The MIVOT viewer generates the model view of the data
m_viewer = MivotViewer(query_result, resolve_ref=True)

# Build a Python object matching the TEMPLATES content and
# which leaves are set with the values of the first row
mango_property = m_viewer.dm_instance
```

Get SkyCoord Object through MIVOT/MANGO

```
# Enable MIVOT-specific features in the pyvo library
activate_features("MIVOT")

scs_srv = SCSService("https://vizier.cds.unistra.fr/viz-bin/conesearch/V1.5/I

query_result = scs_srv.search(
    pos=SkyCoord(ra=52.26708 * u.degree, dec=59.94027 * u.degree, frame='icrs
    radius=0.5)

# The MIVOT viewer generates the model view of the data
m_viewer = MivotViewer(query_result, resolve_ref=True)
```

```
# Build a Python object matching the TEMPLATES content and
# which leaves are set with the values of the first row
mango_property = m_viewer.dm_instance
```

```
while m_viewer.next_row_view():
    if mango_property.dmtypes == "mango:EpochPosition":
        scb = SkyCoordBuilder(mango_property.to_dict())
        # do whatever process with the SkyCoord object
        print(scb.build_sky_coord())
```

Get SkyCoord Object through MIVOT/MANGO

```
# Enable MIVOT-specific features in the pyvo library
activate_features("MIVOT")
```

```
scs_srv = SCSService("https://vizier.cds.unistra.fr/viz-bin/conesearch/V1.5/I
```

```
query_result = scs_srv.search(
    pos=SkyCoord(ra=52.26708 * u.degree, dec=59.94027 * u.degree,
    radius=0.5)
```

```
# The MIVOT viewer generates the model view of the data
m_viewer = MivotViewer(query_result, resolve_ref=True)
```

```
# Build a Python object matching the TEMPLATES content and
# which leaves are set with the values of the first row
mango_property = m_viewer.dm_instance
```

```
while m_viewer.next_row_view():
    if mango_property.dctype == "mango:EpochPosition":
        scb = SkyCoordBuilder(mango_property.to_dict())
        # do whatever process with the SkyCoord object
        print(scb.build_sky_coord())
```

```
<SkyCoord (ICRS): (ra, dec, distance) in (deg, deg, pc)
(51.64272638, 60.28156089, 5119.99988556)
(pm_ra_cosdec, pm_dec) in mas / yr
(13.31, -23.43)>
<SkyCoord (ICRS): (ra, dec, distance) in (deg, deg, pc)
(52.0981839, 60.2557263, 3799.99995232)
(pm_ra_cosdec, pm_dec) in mas / yr
(23.1, -22.31)>
<SkyCoord (ICRS): (ra, dec, distance) in (deg, deg, pc)
(52.12106121, 59.90665623, 920.00001669)
(pm_ra_cosdec, pm_dec) in mas / yr
(-0.56, -3.62)>
<SkyCoord (ICRS): (ra, dec, distance) in (deg, deg, pc)
(52.26722684, 59.94033461, 759.99999046)
(pm_ra_cosdec, pm_dec) in mas / yr
(-0.82, -1.85)>
```

Get SkyCoord Object through MIVOT/MANGO

```
# Enable MIVOT-specific features in the pyvo library
activate_features("MIVOT")

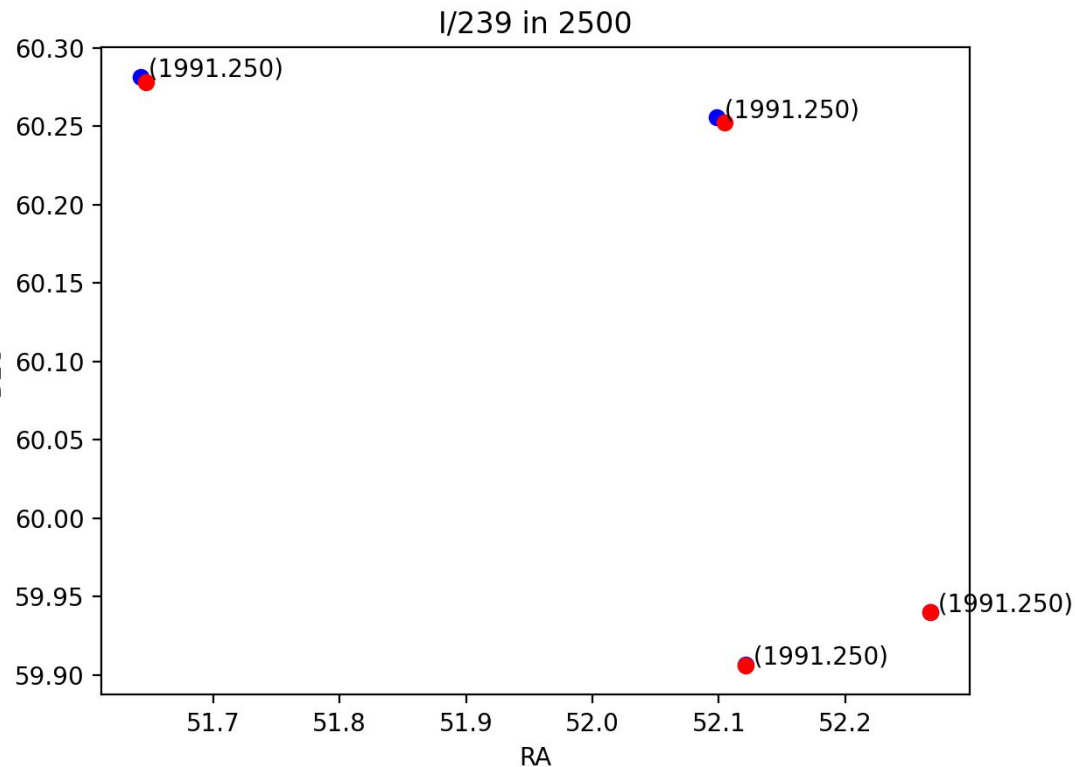
scs_srv = SCSService("https://vizier.cds.unistra.fr/vi

query_result = scs_srv.search(
    pos=SkyCoord(ra=52.26708 * u.degree, dec=59.94027
    radius=0.5)

# The MIVOT viewer generates the model view of the data
m_viewer = MivotViewer(query_result, resolve_ref=True)
```

```
# Build a Python object matching the TEMPLATES content
# which leaves are set with the values of the first row
mango_property = m_viewer.dm_instance
```

```
while m_viewer.next_row_view():
    if mango_property.dmtypes == "mango:EpochPosition":
        scb = SkyCoordBuilder(mango_property.to_dict())
        # do whatever process with the SkyCoord object
        print(scb.build_sky_coord())
```



Get SkyCoord Object through MIVOT/MANGO

```
# Enable MIVOT-specific features in the pyvo library
activate_features("MIVOT")

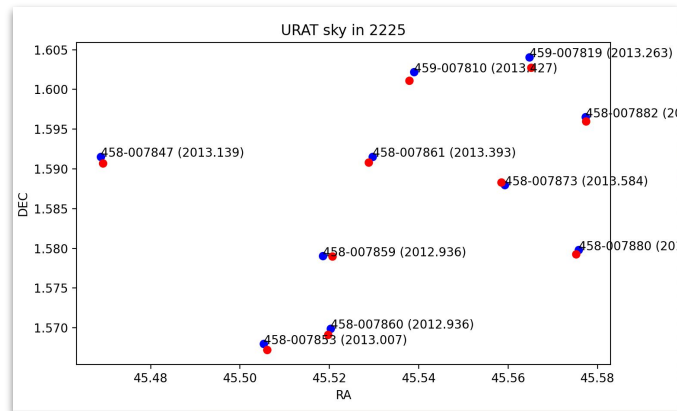
scs_srv = SCSService("https://vizier.cds.unistra.fr/viz-bin/conesearch/V1.5/I

query_result = scs_srv.search(
    pos=SkyCoord(ra=52.26708 * u.degree, dec=59.94027 * u.degree, frame='icrs'
    radius=0.5)

# The MIVOT viewer generates the model view of the data
m_viewer = MivotViewer(query_result, resolve_ref=True)
```

```
# Build a Python object matching the TEMPLATES content and
# which leaves are set with the values of the first row
mango_property = m_viewer.dm_instance
```

```
while m_viewer.next_row_view():
    if mango_property.dctype == "mango:EpochPosition":
        scb = SkyCoordBuilder(mango_property.to_dict())
        # do whatever process with the SkyCoord object
        print(scb.build_sky_coord())
```



This code is not specific to a particular VOTable.

The processing run in the loop can be applied to any annotated query result.

Build MIVOT Annotations with PyVO (1.7+)

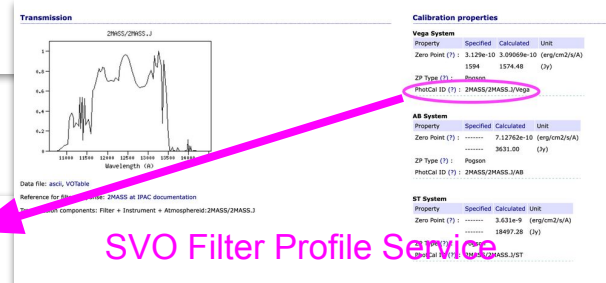
```
votable = parse(votable_path)
builder = InstancesFromModels(votable, dmid="URAT1")
```

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

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

Example: 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"})
```

SVO Filter Profile Service

A Quick Look at the MIVOT Block

```
<VODML xmlns="http://www.ivoa.net/xml/mivot">
  <REPORT status="OK"/>
  <MODEL name="ivoa" url="https://www.ivoa.net/xml/VODML/IVOA-v1.vo-dml.xml"/>
  <MODEL name="mango"
    url="https://raw.githubusercontent.com/ivoa-std/MANGO/refs/heads/wd-v1.0/vo-dml/mango.vo-dml.xml"/>
  <MODEL name="Phot" url="https://ivoa.net/xml/VODML/Phot-v1.vodml.xml"/>
  <GLOBALS>
    <INSTANCE dmtpe="Phot:PhotCal" dmid="_photcal_2MASS_2MASS_J_AB">
    <INSTANCE dmtpe="Phot:PhotometryFilter" dmid="_photfilter_2MASS_2MASS_J_AB">
  </GLOBALS>
  <TEMPLATES>
    <INSTANCE dmtpe="mango:MangoObject" dmid="URAT1">
      <ATTRIBUTE dmtpe="ivoa:string" dmrole="mango:MangoObject.identifier" ref="URAT1"/>
      <COLLECTION dmrole="mango:MangoObject.propertyDock">
        <INSTANCE dmtpe="mango:Brightness">
          <ATTRIBUTE dmtpe="ivoa:string" dmrole="mango:Property.description" value="magnitude J"/>
          <INSTANCE dmtpe="mango:VocabularyTerm" dmrole="mango:Property.semantics">
            <ATTRIBUTE dmtpe="ivoa:string" dmrole="mango:VocabularyTerm.uri"
              value="https://www.ivoa.net/rdf/uat/2024-06-25/uat.html#magnitude"/>
            <ATTRIBUTE dmtpe="ivoa:string" dmrole="mango:VocabularyTerm.label" value="magnitude"/>
          </INSTANCE>
          <ATTRIBUTE dmtpe="ivoa:RealQuantity" dmrole="mango:PhotometricProperty.value" unit="mag" ref="Jmag"/>
          <INSTANCE dmtpe="mango:error.PErrorSymID" dmrole="mango:PhotometricProperty.error">
            <ATTRIBUTE dmtpe="ivoa:RealQuantity" dmrole="mango:error.PErrorSymID.sigma" unit="mag" ref="e_Jmag"/>
          </INSTANCE>
          <REFERENCE dmrole="mango:Brightness.photCal" dmref="_photcal_2MASS_2MASS_J_AB"/>
        </INSTANCE>
      </COLLECTION>
    </INSTANCE>
  </TEMPLATES>
</VODML>
```

Read the Mapped Magnitude

```
m_viewer = MivotViewer(votable, resolve_ref=True)
mivot_instance = m_viewer.dm_instance
while m_viewer.next():
    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":
                mag_value = mango_property.value.value
                mag_error = mango_property.error.sigma.value
                mag_wl = mango_property.photCal.photometryFilter.spectralLocation.value.value
                mag_filter = mango_property.photCal.identifier.value
                print(f"magnitude at {mag_wl} Angstrom (filter {mag_filter}) is {mag_value:.2f} +/- {mag_error:.2f}")
        break
```

Read source 458-007880

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

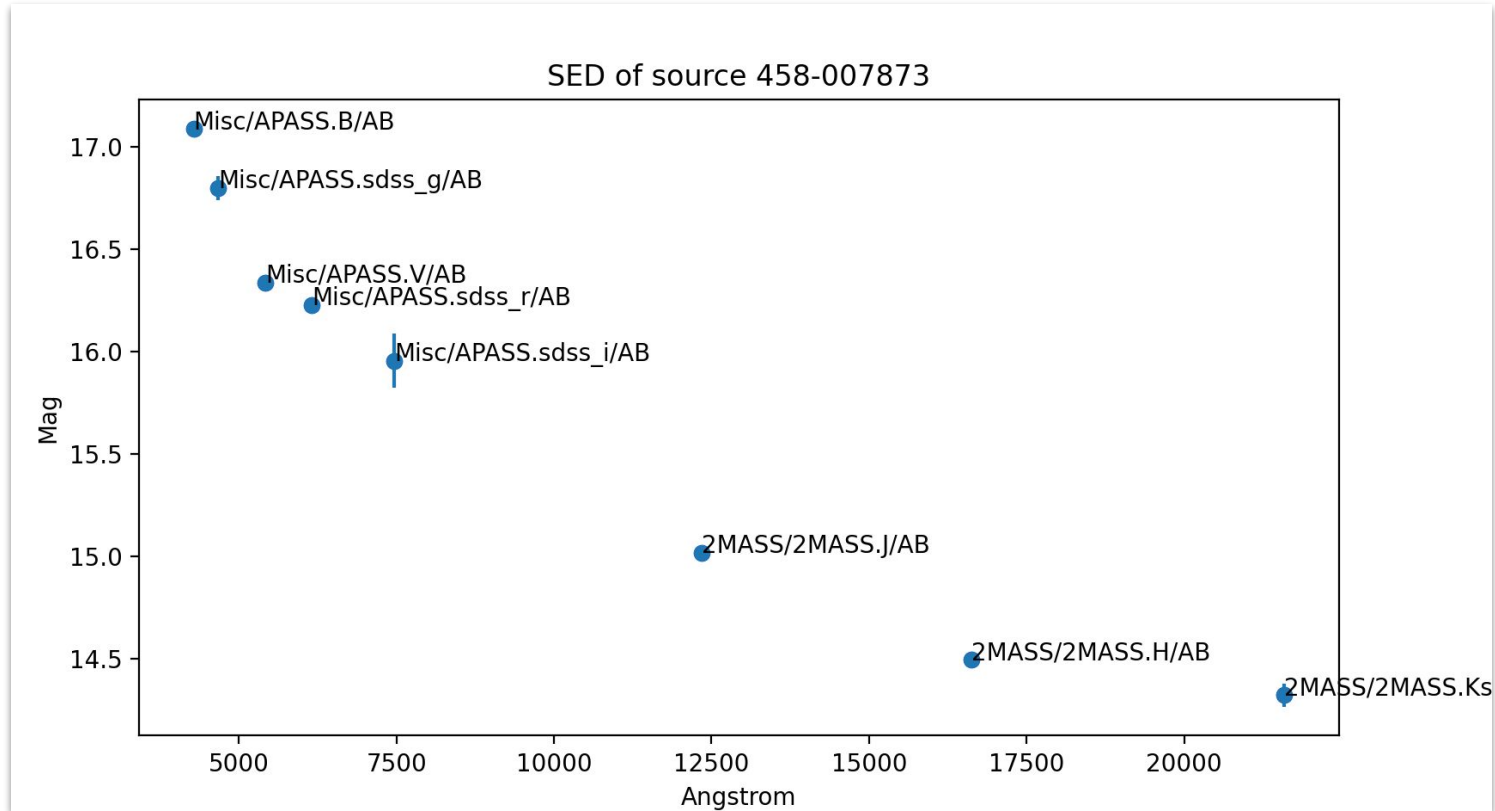
Let's do the Same for all Bands

```
<VODML xmlns="http://www.ivoa.net/xml/mivot">
  <REPORT status="OK"/>
  <MODEL name="ivoa" url="https://www.ivoa.net/xml/VODML/IVOA-v1.vo-dml.xml"/>
  <MODEL name="mango"
    url="https://raw.githubusercontent.com/ivoa-std/MANGO/refs/heads/wd-v1.0/vo-dml/mango.vo-dml.xml"/>
  <MODEL name="Phot" url="https://ivoa.net/xml/VODML/Phot-v1.vodml.xml"/>
  <GLOBALS>
    <INSTANCE dmttype="Phot:PhotCal" dmid="_photcal_2MASS_2MASS_J_AB">
    <INSTANCE dmttype="Phot:PhotometryFilter" dmid="_photfilter_2MASS_2MASS_J_AB">
    <INSTANCE dmttype="Phot:PhotCal" dmid="_photcal_2MASS_2MASS_H_AB">
    <INSTANCE dmttype="Phot:PhotometryFilter" dmid="_photfilter_2MASS_2MASS_H_AB">
    <INSTANCE dmttype="Phot:PhotCal" dmid="_photcal_2MASS_2MASS_Ks_AB">
    <INSTANCE dmttype="Phot:PhotometryFilter" dmid="_photfilter_2MASS_2MASS_Ks_AB">
    <INSTANCE dmttype="Phot:PhotCal" dmid="_photcal_Misc_APASS_B_AB">
    <INSTANCE dmttype="Phot:PhotometryFilter" dmid="_photfilter_Misc_APASS_B_AB">
    <INSTANCE dmttype="Phot:PhotCal" dmid="_photcal_Misc_APASS_V_AB">
    <INSTANCE dmttype="Phot:PhotometryFilter" dmid="_photfilter_Misc_APASS_V_AB">
    <INSTANCE dmttype="Phot:PhotCal" dmid="_photcal_Misc_APASS_sdss_g_AB">
    <INSTANCE dmttype="Phot:PhotometryFilter" dmid="_photfilter_Misc_APASS_sdss_g_AB">
    <INSTANCE dmttype="Phot:PhotCal" dmid="_photcal_Misc_APASS_sdss_r_AB">
    <INSTANCE dmttype="Phot:PhotometryFilter" dmid="_photfilter_Misc_APASS_sdss_r_AB">
    <INSTANCE dmttype="Phot:PhotCal" dmid="_photcal_Misc_APASS_sdss_i_AB">
    <INSTANCE dmttype="Phot:PhotometryFilter" dmid="_photfilter_Misc_APASS_sdss_i_AB">
  </GLOBALS>
  <TEMPLATES>
    <INSTANCE dmttype="mango:MangoObject" dmid="URAT1">
      <ATTRIBUTE dmttype="ivoa:string" dmrole="mango:MangoObject.identifier" ref="URAT1"/>
      <COLLECTION dmrole="mango:MangoObject.propertyDock">
        <INSTANCE dmttype="mango:Brightness">
        <INSTANCE dmttype="mango:Brightness">
        <INSTANCE dmttype="mango:Brightness">
        <INSTANCE dmttype="mango:Brightness">
        <INSTANCE dmttype="mango:Brightness">
        <INSTANCE dmttype="mango:Brightness">
        <INSTANCE dmttype="mango:Brightness">
      </COLLECTION>
    </INSTANCE>
  </TEMPLATES>
</VODML>
```

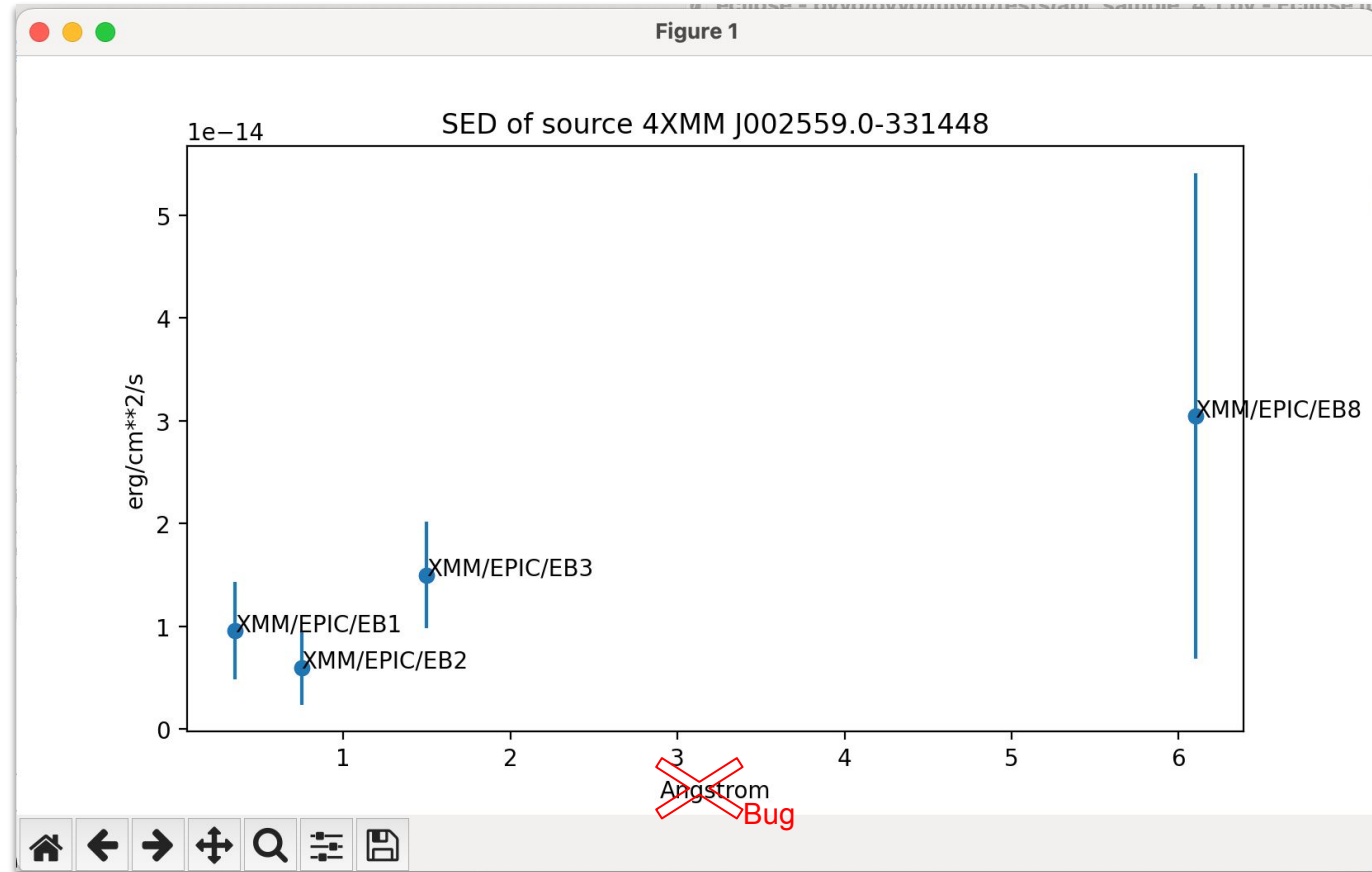
Read all Bands

```
for mango_property in mivot_instance.propertyDock:
    if mango_property.dmttype == "mango:Brightness":
        if mango_property.value.value:
            mag.append(mango_property.value.value)
            mag_error.append(mango_property.error.sigma.value)
            sp_location.append(
                mango_property.photCal.photometryFilter.spectralLocation.value.value)
            sp_filter.append(mango_property.photCal.identifier.value)
plot_sed(mivot_instance.identifier.value, mag, mag_error, sp_filter, sp_location)
```

And Plot Them



The same code applied to XMM data (xtdb)



VALIDATOR

🏠 mivot-validator

latest

Search docs

CONTENTS:

Welcome to mivot-validator's documentation!

Indices and tables

mivot_validator package

mivot_validator.instance_checking package

mivot_validator.instance_checking.builder package

mivot_validator.instance_checking.xml_inter package

mivot_validator.launchers package

mivot_validator.utils package

mivot_validator

TIDELIFT

The 2024 Tidelift Open Source Sustainability Report

Read the report

Hot off the press! The 2024 Tidelift state of the OSS maintainer report.

Read now!

Ad by EthicalAds

🏠 / Welcome to Mivot Validator Tools [Edit on GitHub](#)

Welcome to Mivot Validator Tools

This package has 2 purposes:

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

Python 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

XML Schema Validation

Validation of an annotated VOTable against both VOTable and MIVOT schemas:

```
$ mivot-validate PROJECT_DIR/tests/data/gaia_3mags_ok_1.xml

USAGE: mivot-validate [path]
Validate against both VOTable and MIVOT schemas
path: either a simple file or a directory
      all directory XML files are validated
exit status: 0 in case of success, 1 otherwise
```

Validation of an annotated VOTable against the MIVOT schema:

```
$ mivot-mapping-validate PROJECT_DIR/tests/data/gaia_3mags_ok_1.xml

USAGE: mivot-mapping-validate [path]
Validate XML files against MIVOT schema
path: either a simple file or a directory
      all directory XML files are validated
exit status: 0 in case of success, 1 otherwise
```

Search projects

Help Sponsors Log in Register

mivot-validator 0.2

Latest version

Released: Oct 3, 2024

`pip install mivot-validator`

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**

Maintainers

[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

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 instantiated

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

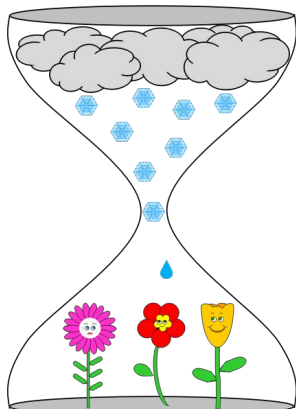
1. Child of INSTANCE
The ATTRIBUTE can be seen as a class attribute; it must have a @drole XML attribute.
In this case, the ATTRIBUTE must be specified by:
 - * @ref - reference to a VOTable PARAM or FIELD
 - * @value - a literal
 - * if both are provided; @value serves as the default if the reference 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:

STATUS AND PROSPECTS



<https://pyvo.readthedocs.io/en/latest/mivot/>

Waiting MANGO is a REC before to extend the PyVO implementation

- MANGO-centric feature
- More robust

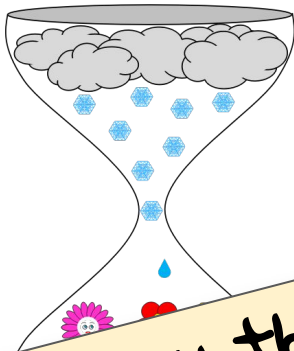
Available in Pyvo

- Viewer: 1.6+
- Writer: 1.7+

Hope to have new implementations

- services
- client codes

STATUS AND PROSPECTS



<https://pyvo.readthedocs.io/en/latest/>

[mivot/](https://pyvo.readthedocs.io/en/latest/mivot/)

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

Waiting for more to
external implementation

- More PyVO-centric feature
- More robust

Available in Pyvo

- Reader: 1.6+
- Writer: 1.7+

Hope to have new implementations

- services
- client codes

backup

MODEL

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

SERVER SIDE

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

CLIENT SIDE

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

VALIDATOR

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

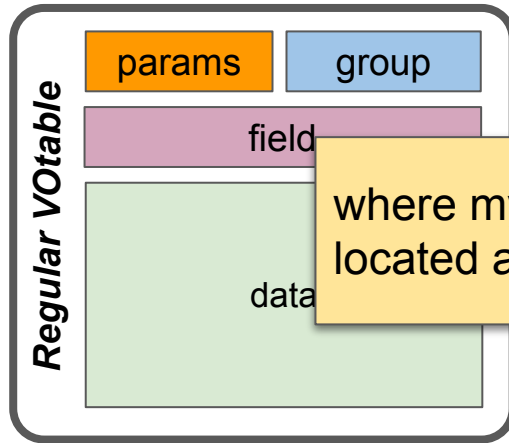
ONLINE DOCUMENTATION

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

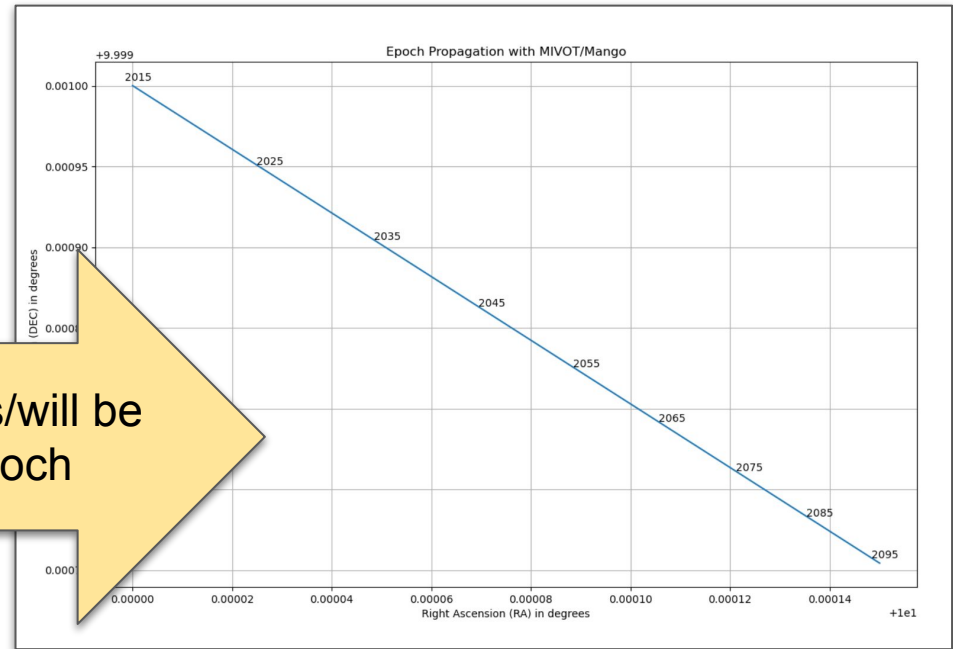
MISCELLANEOUS

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

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

Legacy: Limitation of the COOSYS solutions



- The `@ref` semantic or **role** is **not well defined**
- The way fields connected to coosys **do interact** together is **poorly defined**
- **No** convenient way to show up **errors with correlations**
- According the the VOTable document, solution based on coosys are **short term solution**

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.

- We **miss a model** showing how all of these **parameters** do **work together**.

MIVOT: The Model Based Alternative

- 0 Keep the VOTable as delivered by the DAL engine**
 - No FIELDREF tuning
 - No GROUP to create
- 1 Use a VO model describing the EPOCH propagation**
 - 6 parameters
 - Errors
 - Space Frame
 - Time System
- 2 Write an XML serialization of that MODEL in the VOTable**
 - MIVOT syntax
- 3 Insert that XML piece above the TABLE**
 - Put the reference of the matching FIELDS into the model leaves

IVOA Recommendation

IVOA Documents



IVOA Recommendation 20 June 2023

Interest/Working Group:

<http://www.ivoa.net/twiki/bin/view/IVOA/IvoaDataModel>

Author(s):

Laurent Michel, Mark Cresitello-Dittmar, François Bonnarel, Gilles La

Editor(s):

Laurent Michel, Mark Cresitello-Dittmar

Abstract

Abstract: Model Instances in VOTables (MIVOT) defines a syntax to map VOTable data to the data model elements (class, attributes, types, etc.) of a standardized metadata language. The data model elements are grouped in an independent hierarchy. The MIVOT syntax allows to describe a data structure as a hierarchy of classes. It also allows to describe missing data. Missing metadata can also be provided using MIVOT, for instance by providing information about the client and server sides. The adopted design does not alter the original VOTable format.

Status of this document

This document has been produced by the Data Model Working Group.

It has been reviewed by IVOA Members and other interested parties, and has been reference from another document. IVOA's role in making the Recommendation is the Community.

Available formats: pdf

- An IVOA standard document
- An XML schema allowing computer to validate documents.
 - The schema has been written in XSD1.1 to support different syntax patterns depending on the local context
 - The schema is independent from the VOTable: tools not supporting MIVOT are still working on annotated VOTables

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();

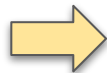
// Retrieve the EpochPosition instance from the mapping block
let epoch_positions = model_layer.get_epoch_positions_as_ref().get("EpochPosition").unwrap();

// 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

MODEL

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

SERVER SIDE

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

CLIENT SIDE

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