



# MANGO

## Model for ANnotating Generic Objects

Laurent MICHEL, F. Bonnarel, G.Landais, M. Louys, M. Molinaro, J. Salgado



Thodore Rosswith, Pine.

Gabriel Sculp.

Thodore Rosswith, Pine.

Gabriel Sculp.

Thodore Rosswith, Pine.

Gabriel Sculp.

MANGUIER ou MANGO.

MANGUIER ou MANGO.

MANGUIER ou MANGO.

# Reading VOTables Quantity per Quantity

```

<RESOURCE type="meta"><URL url="http://www.ivoa.net/xml/ivot/" />
<REPORT status="OK" />
<MODEL name="vot" />
<GLOBAL name="vot" url="https://www.ivoa.net/xml/VODML/IVOA-v1-vo-dml.xml"/>
<GLOBAL name="vot" />
<INSTANCE dsi="coords:SpaceSys" dsid="SpaceFrame_ICRS"/>
<ATTRIBUTE dsi="coords:SpaceFrame" dsi="SpaceFrame" value="ICRS"/>
</INSTANCE>
</GLOBAL>
<PLATES labelref="I/229/urat1">
<INSTANCE dsi="mango:EpochPosition">
<ATTRIBUTE dsi="mango:EpochPosition.longitude" dsi="ivoa:RealQuantity" ref="RAEPOCH" unit="deg"/>
<ATTRIBUTE dsi="mango:EpochPosition.latitude" dsi="ivoa:RealQuantity" ref="DECPOCH" unit="deg"/>
<ATTRIBUTE dsi="mango:EpochPosition.pmLongitude" dsi="ivoa:RealQuantity" ref="pmRA" unit="mas/y"/>
<ATTRIBUTE dsi="mango:EpochPosition.pmLatitude" dsi="ivoa:RealQuantity" ref="pmDEC" unit="mas/y"/>
</INSTANCE>
</PLATES>
</VODML></RESOURCE>
    
```

Browse all available available quantities

Select quantities of interest

How to reconstruct complex quantities is described in the VOTable itself

**Complex quantities** associating several columns are described in the VOTable by the data provider.

- The syntax is defined by the MIVOT standard (apps1)
- The mapped model is MANGO

Construct the scientific code from these quantities

```

<DESCRIPTION>(mag) mean IBAT nodes IJ magnitude (4)/</DESCRIPTION>
</GLOBAL>
</INSTANCE>
</PLATES>
</VODML></RESOURCE>
    
```

Table <FIELDS>

```

coordinate(
    complex_quantity.longitude.value,
    latitude=complex_quantity.latitude.value,
    pmLongitude=complex_quantity.pmLongitude.value,
    pmLatitude=complex_quantity.pmLatitude.value,
    frame=mivot_object.Coordinate_coordSys.spaceRefFrame.value,
)
    
```

# What Are Complex Quantities (or Property)

Quantities with <b>more than one coordinates</b>	<ul style="list-style-type: none"><li>● Position</li><li>● proper motion</li><li>● CCD position</li><li>● errors</li></ul>
Quantity with <b>errors</b>	<ul style="list-style-type: none"><li>● many</li></ul>
Quantity with <b>specific coordinate systems</b>	<ul style="list-style-type: none"><li>● Photometry</li><li>● Moving objects (space + time)</li></ul>
Quantity <b>linked</b> with <b>other quantities</b>	<ul style="list-style-type: none"><li>● Photometry + time stamps</li><li>● Position + quality flag</li></ul>
<b>Correlated</b> quantities	<ul style="list-style-type: none"><li>● Position + proper motion</li></ul>
<b>Mix</b> of all above cases	

# The MANGO Project

## The Concept

- A flexible **model** helping to interpret the **content** of (m)any **data tables**
- Make many things **machine-readable**
- A data model that is intended for use in conjunction with MIVOT to annotate **DAL query responses**

## History

- Born after a poll to get **use-cases** for a **source data-model** (Paris 2019)
- Kept dormant while the pandemic, the DM workshop and the MIVOT process

## Watchout

- MANGO is **not** a model describing any particular **science product**.

# Design Guideline

## **Flexible enough to cover most of the use cases**

- Applicable for legacy or mission data
- Coverage of various domains (HE, time domain...)

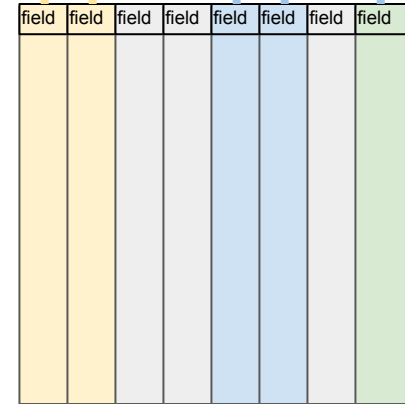
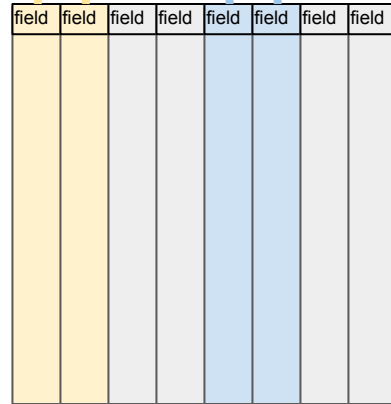
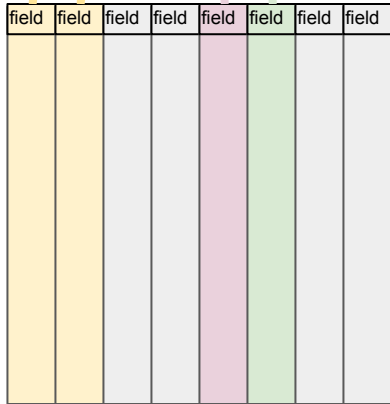
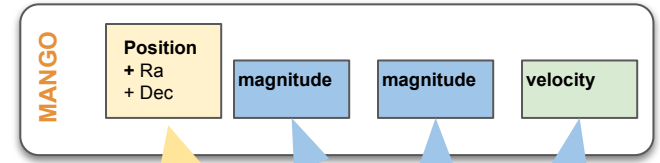
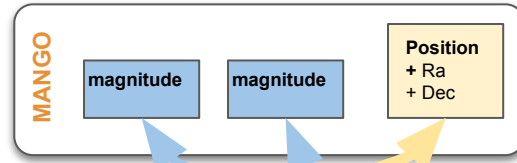
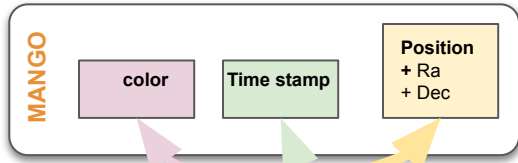
## **Simple Enough to be comfortable with the mapping**

- Mapping easy to build
- Mapping easy to consume
- Object hierarchy as shallow as possible

## **Accurate enough to provide significant added value**

- Domains not covered before
- Property association
- Data correlation

# The Same Model for Different Datasets

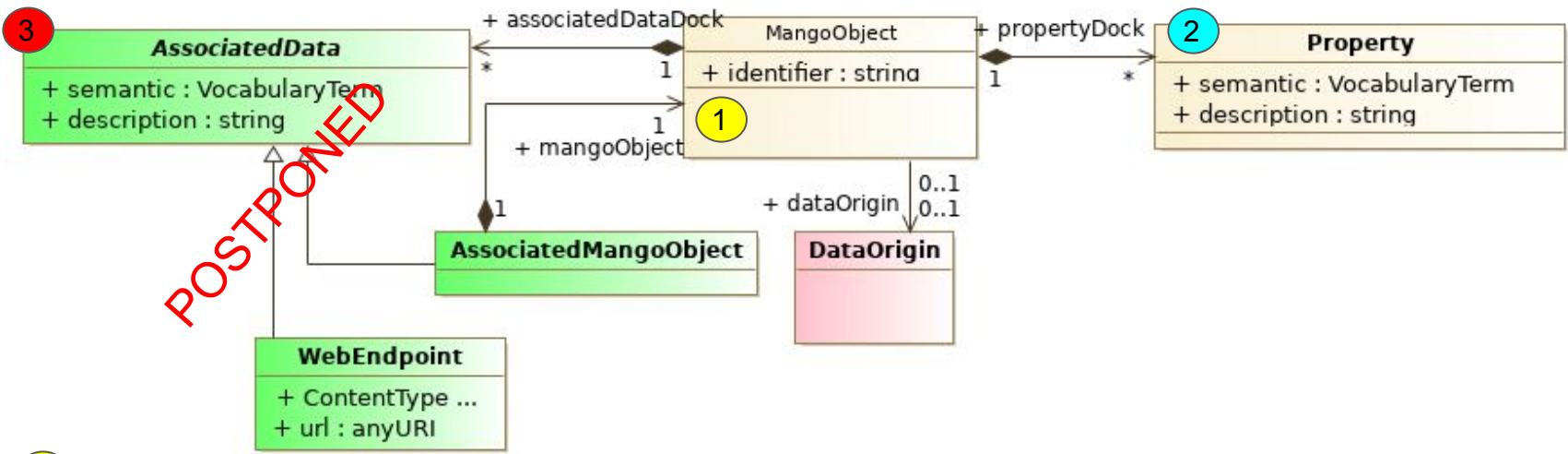


## The model describes

- Global meta data
- The quantity container
- Individual quantities
- Quantity associations

**The model does not specify any pattern for the expected quantities.**

# Model Overview

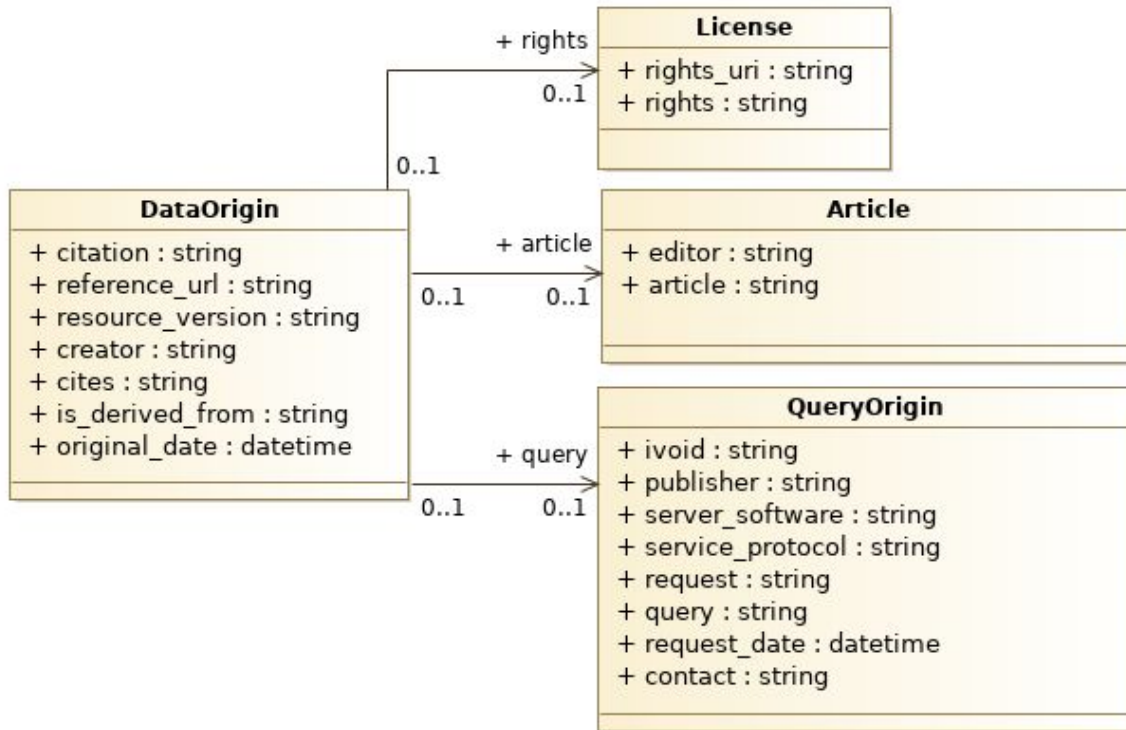


- 1
- ### Model Core
- Identifier
  - Data origin

- 2
- ### One dock for the Properties
- EpochPosition
  - Magnitude
  - Color
  - Radial velocity
  - Desired metadata

- 3
- ### One dock for the ass. data
- Links on any dataset or service
  - Links to other Mango objects

# The Data Origin (DCP IG)



## Trace the dataset origin

- Article
- License
- Query
- Not the provenance

## Some of this information is in <INFO>

- The model provides a structured view on it

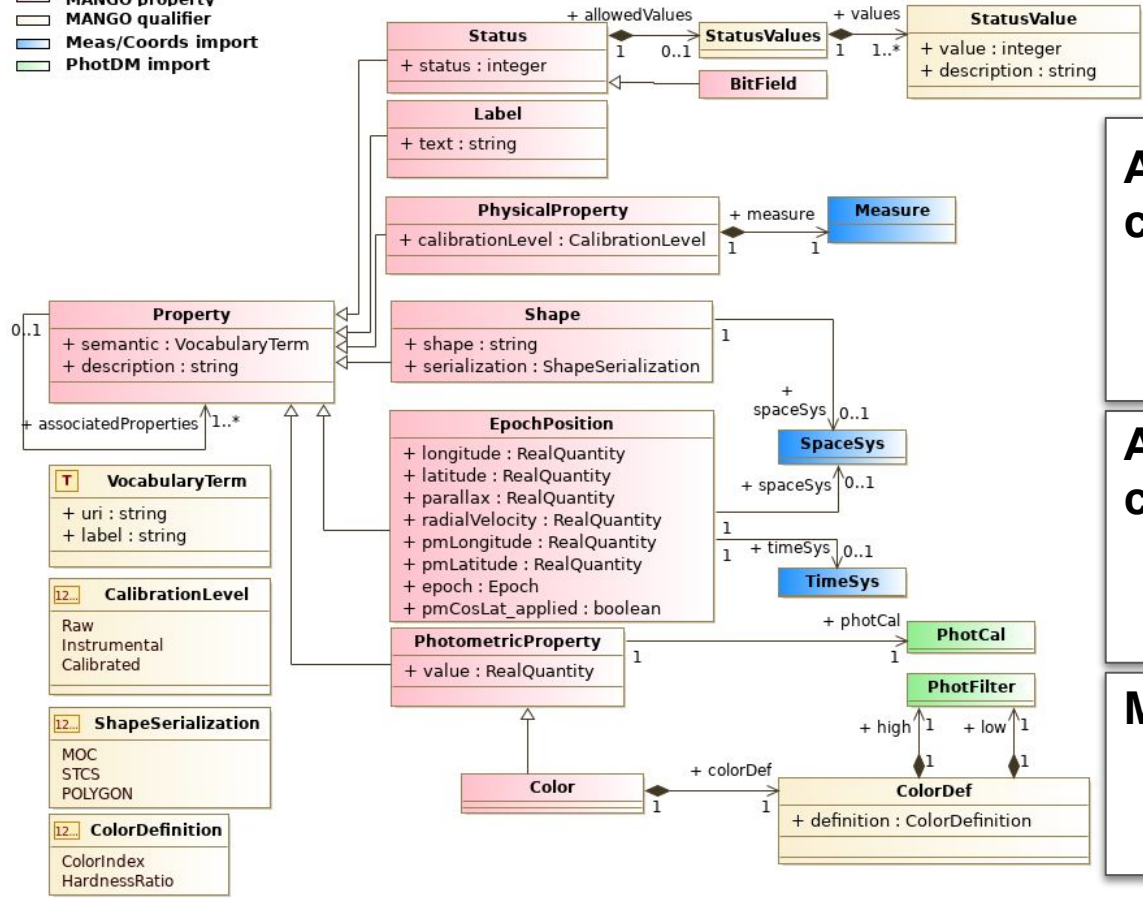
## TBD - TBC

- Merge with or reuse DatasetDm



# Properties Supported by the Model

- MANGO property
- MANGO qualifier
- Meas/Coords import
- PhotDM import



**A mix of built-in and imported classes**

- Measurement
- Coordinates
- PhotDM

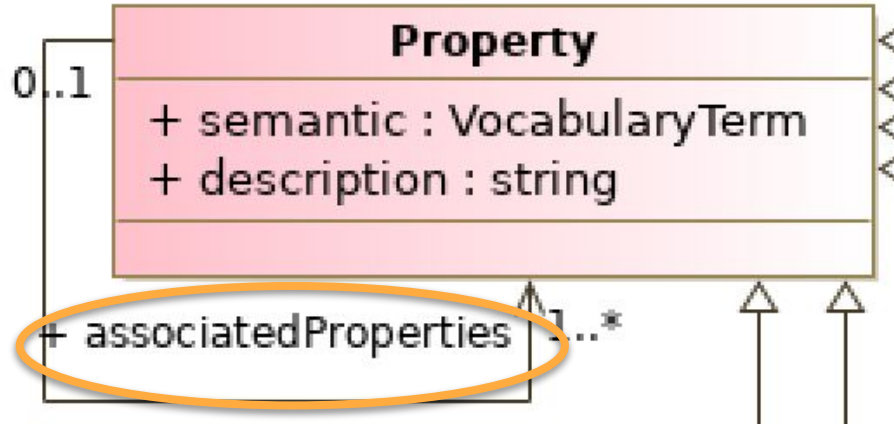
**A limited number of specific classes**

- Covering the most popular use-cases
- Can be still extended

**Make things machine-readable**

- Associations with coord. systems
- Serialization modes
- Computation mode

# Property Semantic



## Property role refined by semantic tags

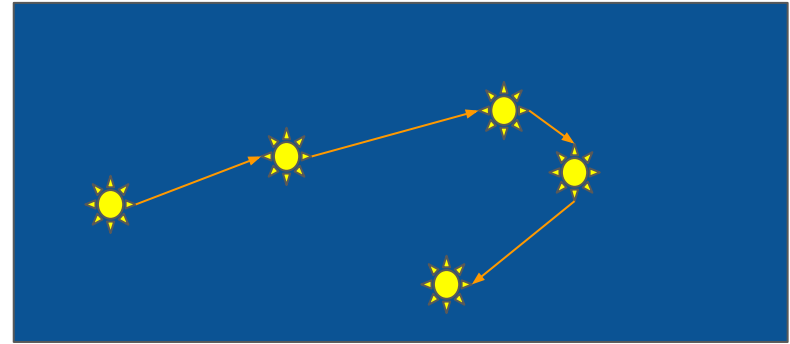
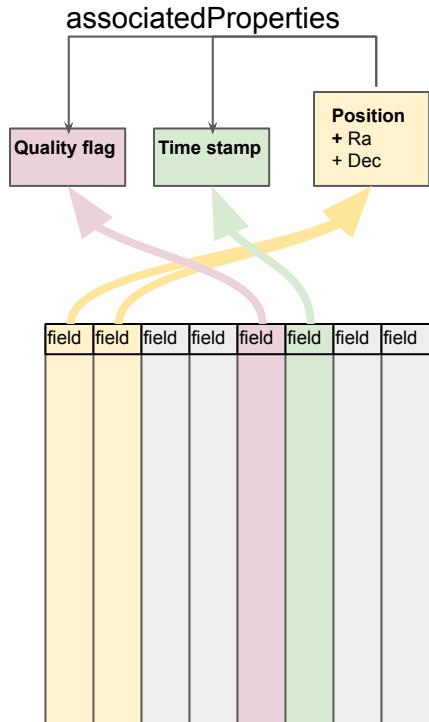
- Vocabulary
- Free text description

## Semantic block can relate to

- The property itself
- The property + the associated properties

Property	Description
<b>PhotometricProperty</b>	This is a V magnitude
<b>Time</b>	This is a time stamp
<b>PhotometricProperty + Time</b>	This is a photometric point in V band

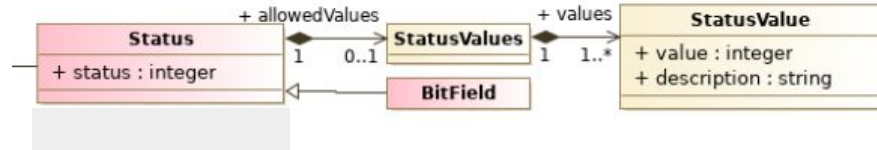
# Property Association



## Association purpose

- Associating a property with a time makes it easier to interpret time-domain data.
- Associating a property with a quality flag makes it easier to filter data.

# Status



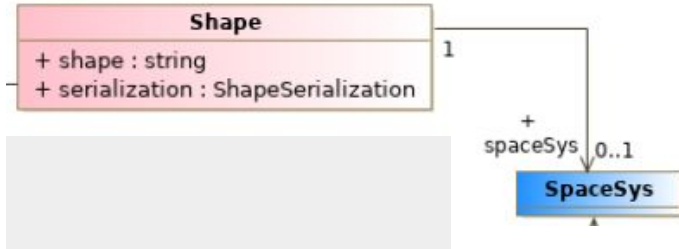
## Make status values machine-readable

- Query setup
- Value understandable by the client

## Set of allowed values

- Allowed values come with their descriptions

# Shape



## String serialization of complex shapes

- Serialization mode (MOC, STC-S...) given by the **ShapeSerialization** enum
- Space coordinate systems imported from **Coordinates** data model

# Support of Classes of the Measurement Model



## Placeholder for classes of the Measurements model

- Position
- Proper Motion
- Velocity
- Time
- Polarization
- Generic Measure

## Structure of the Measure sub-classes

- A value (can be a vector)
- A coordinate system (frame + axis)
- An error

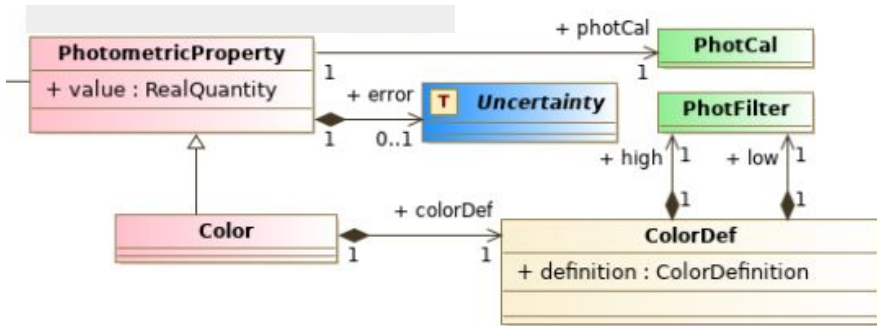
## Great flexibility

- Accurate description of all axis
- Might be tricky to use due to the too many abstractions

## Add a calibration level

- borrowed from Obscore

# Photometry Properties



## Flux/Magnitude

- A simple value with an error and a photometric calibration (imported from `PhotDM`)

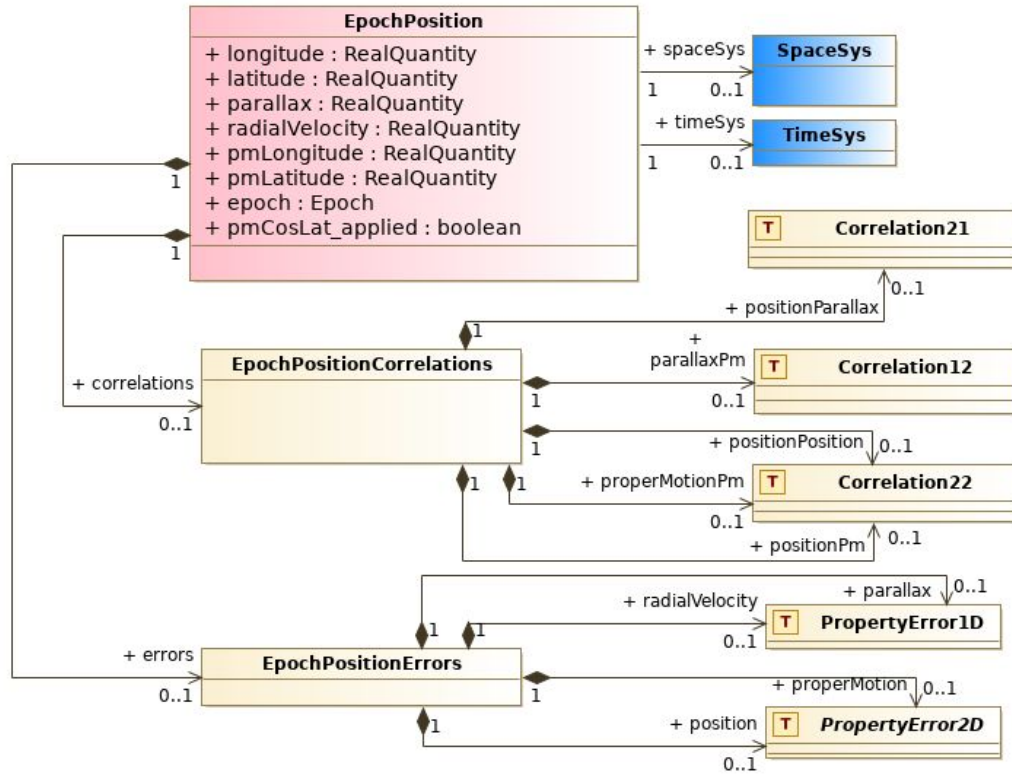
## Color

- A simple value with an error and 2 photometric filters (imported from `PhotDM`)
- Distinction between Color and HR made at model level (`ColorDefinition`)

## TBD - TBC

- Should we make a distinction between Fluxes and Magnitudes at model level?

# The Epoch Position Class



## Inspired by Gaia DR3

### All components in one flat class

- Share the same coordinate systems
- Coordinate systems imported from **Coordinates** data model

### Correlation support

- Correlation matrix split in 1 to 1 sub-matrices
- A 6x6 matrix would be out of control

### Implementations

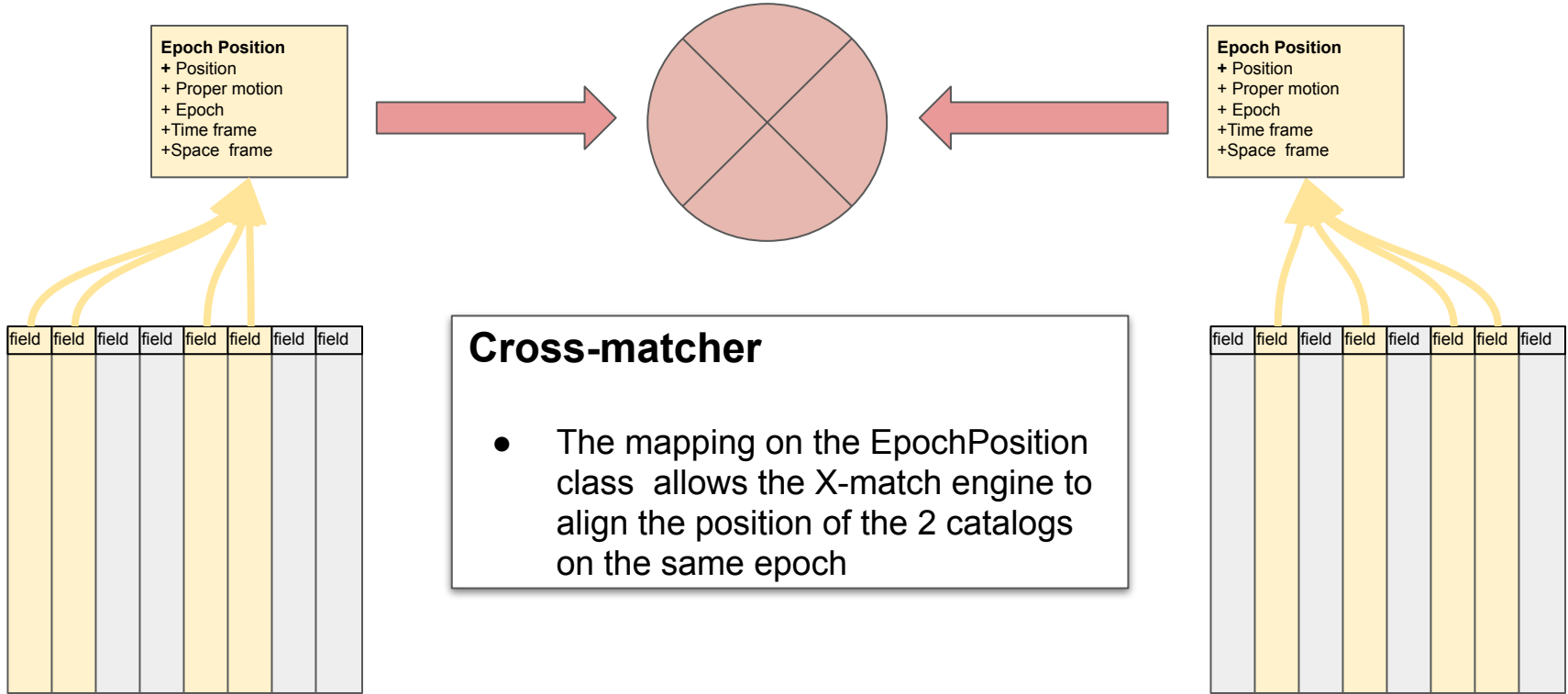
- Vizier cone-search
- Pyvo MIVOT package

### TBD - TBC

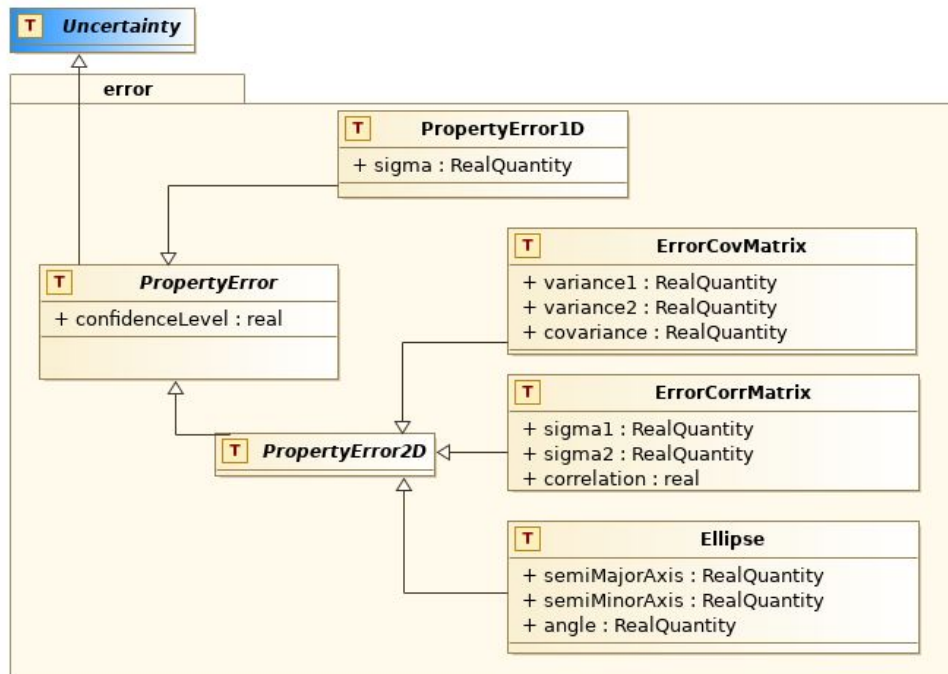
- Need for a validation by stakeholders



# Cross-match Example



# Error Package



Currently designed for the  
**EpochPosition (GAIA)** use-case

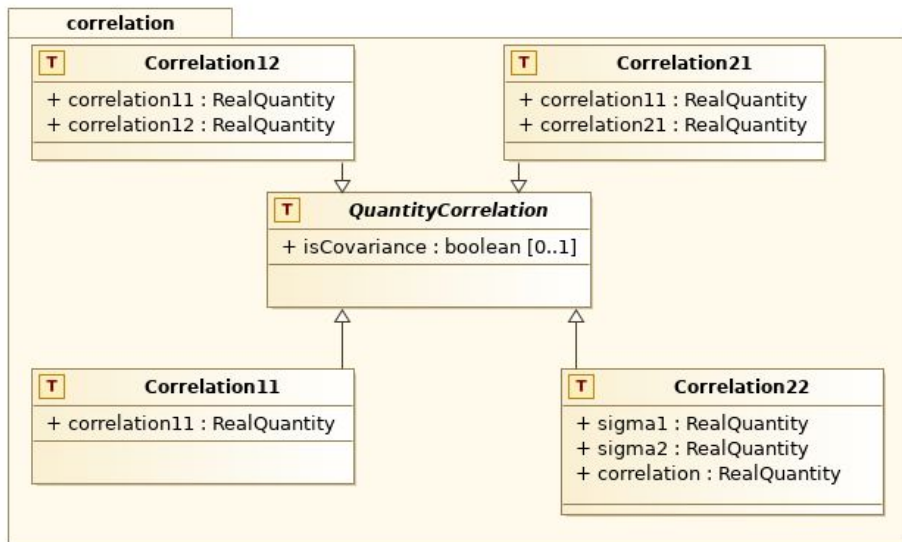
## Main Features

- Confidence level (important for X-Match)
- Split by dimension (1D vs 2D)
- Distinction between covariance and correlation
- Derive from `meas:Uncertainty` to make them usable in the context of the physical properties

## TBD - TBC

- Reuse more **Measurement** model concepts to extend **PropertyError1D**

# Correlation Package



**Currently designed for the  
EpochPosition (GAIA) use-case**

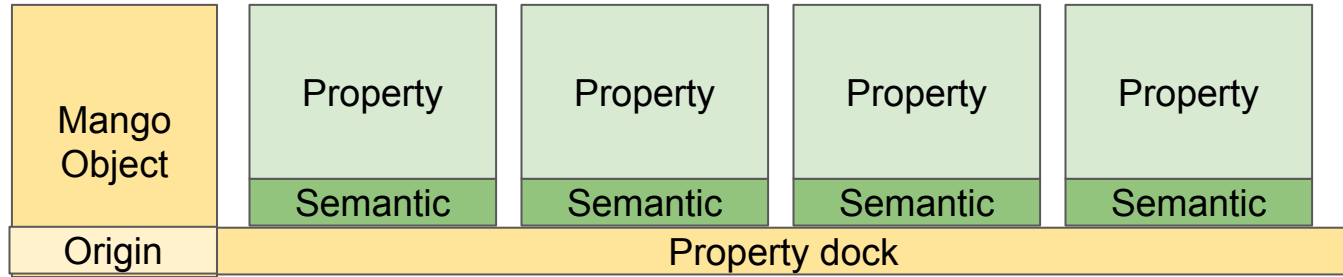
## Main Features

- Support both correlation and covariance
- Split by dimension patterns

## TBD - TBC

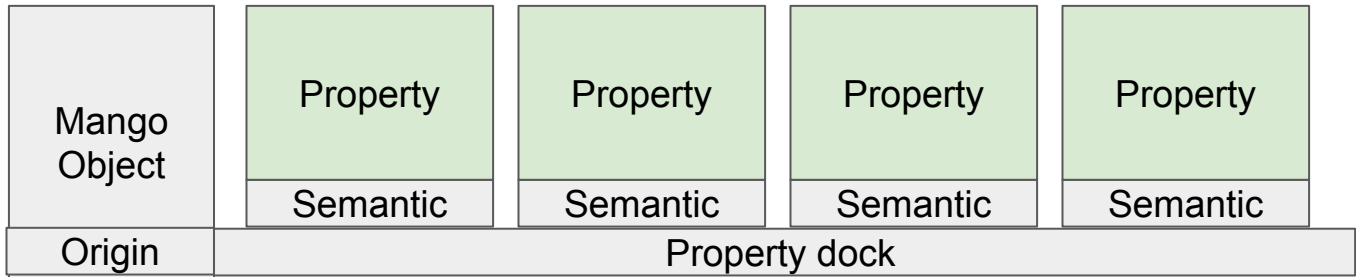
- Must be validated by domain experts
- No use out of the EpochPosition case
- Can be simplified

# Mapping the Whole Model ?



**Data can be mapped as a complete MANGO instance**

# ... or Not



## Or as a bunch of elements as we are doing with the `EpochPosition`

- The parsing remain easy since any component is identified by its `@dmtype` and the `@dmrole` it plays in its context

# Conclusions

**Thanks to MIVOT and PyVO, the model can be exercised against real data.**

- Data sample can be mapped on Mango by hand so that stakeholders can see whether the result match their expectations or not.

**We need expert input**

- Need expert advices to complete the `EpochPosition` class
- Need data provider input to validate the model as a whole
- Need expert in orbiting system: a missing MANGO feature
- Need input for any missing property (redshift, orbiting systems..)

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

**backup**

# Some Open Issues ...

- **Unit modeling**



- 1 **Easier to parse:** no need to search units out of the attributes  
**Not consistent:** accept different units for quantities that must have the same
- 2 **More annoying to deal with:** need to search units as an attribute  
**Consistent:** both quantities always share the same unit

- **How to connect MANGO with IVOA vocabularies**

- IVOA vocabulary (type of associated data)
- XType (shape serialization)
- Imported from other models (ObsCore calibration level)



# A Important Clarification

- **Why some elements present in the VOTable are duplicated in the model?**
  - Property description
  - unit
- **Several (good) Reasons**
  - a. VOTable column descriptions are column related whereas MANGO description are quantity related
  - b. If they are missing in a particular VOTable, we want to be able to set them in the model mapping block.
  - c. We want to be able to export self-consistent model instances
    - i. No longer dependencies with the VOTable context
    - ii. E.g. as JSON feeding a micro-service (see PyVO implementation)

# Reading VOTables Column per Column

```
<FIELD name="l" ucd="pos_eq.ra.main" datatype="double" width="8" precision="8" unit="h" ref="H" ucd="POS_EQ_RA_MAIN" />
<DESCRIPTION>Longitude from center (J2000.0) in hours [ICRS], at Epoch of catalog (epoch)/</DESCRIPTION>
...
<FIELD name="RAICRS" ucd="POS_EQ_RA_MAIN" ref="H" datatype="double" width="11" precision="7" unit="deg" />
<DESCRIPTION>Right ascension on ICRS, at "Epoch" (1)/</DESCRIPTION>
...
<FIELD name="DECICRS" ucd="POS_EQ_DEC_MAIN" ref="H" datatype="double" width="11" precision="7" unit="deg" />
<DESCRIPTION>Declination on ICRS, at "Epoch" (1)/</DESCRIPTION>
...
<FIELD name="pmRA" ucd="pos.pm;pos.eq.ra" ref="H" datatype="float" width="6" precision="1" unit="mas/yr" />
<DESCRIPTION>?(pmr) Proper motion RA*cosDec (from 2MASS) (7)/</DESCRIPTION>
...
<FIELD name="pmDEC" ucd="pos.pm;pos.eq.dec" ref="H" datatype="float" width="6" precision="1" unit="mas/yr" />
<DESCRIPTION>?(pmd) Proper motion in Declination (7)/</DESCRIPTION>
```

Browse all columns to identify their roles

Select columns of interest

```
<FIELD name="RAICRS" ucd="POS_EQ_RA_MAIN" ref="H" datatype="double" width="11" precision="7" unit="deg">
<DESCRIPTION>Right ascension on ICRS, at "Epoch" (1)/</DESCRIPTION>
...
<FIELD name="DECICRS" ucd="POS_EQ_DEC_MAIN" ref="H" datatype="double" width="11" precision="7" unit="deg">
<DESCRIPTION>Declination on ICRS, at "Epoch" (1)/</DESCRIPTION>
...
<FIELD name="pmRA" ucd="pos.pm;pos.eq.ra" ref="H" datatype="float" width="6" precision="1" unit="mas/yr">
<DESCRIPTION>?(pmr) Proper motion RA*cosDec (from 2MASS) (7)/</DESCRIPTION>
...
<FIELD name="pmDEC" ucd="pos.pm;pos.eq.dec" ref="H" datatype="float" width="6" precision="1" unit="mas/yr">
<DESCRIPTION>?(pmd) Proper motion in Declination (7)/</DESCRIPTION>
```

Selected <FIELDS>

Build the scientific code from the selected columns

```
<FIELD name="l" ucd="pos_eq.ra.main" datatype="double" width="8" precision="8" unit="h" ref="H" ucd="POS_EQ_RA_MAIN" />
<DESCRIPTION>Longitude from center (J2000.0) in hours [ICRS], at Epoch of catalog (epoch)/</DESCRIPTION>
...
<FIELD name="RAICRS" ucd="POS_EQ_RA_MAIN" ref="H" datatype="double" width="11" precision="7" unit="deg" />
<DESCRIPTION>Right ascension on ICRS, at "Epoch" (1)/</DESCRIPTION>
...
<FIELD name="DECICRS" ucd="POS_EQ_DEC_MAIN" ref="H" datatype="double" width="11" precision="7" unit="deg" />
<DESCRIPTION>Declination on ICRS, at "Epoch" (1)/</DESCRIPTION>
...
<FIELD name="pmRA" ucd="pos.pm;pos.eq.ra" ref="H" datatype="float" width="6" precision="1" unit="mas/yr" />
<DESCRIPTION>?(pmr) Proper motion RA*cosDec (from 2MASS) (7)/</DESCRIPTION>
...
<FIELD name="pmDEC" ucd="pos.pm;pos.eq.dec" ref="H" datatype="float" width="6" precision="1" unit="mas/yr" />
<DESCRIPTION>?(pmd) Proper motion in Declination (7)/</DESCRIPTION>
```

Quantities associating several columns must be reconstructed by the client code, whereas knowledge of these quantities is held by the data provider.

Table <FIELDS>

```
for rec in table.array:
    skycoord = SkyCoord(longitude=rec["RAICRS"],
                        latitude=rec["DECICRS"],
                        pmLongitude=rec["pmRA"],
                        pmLatitude=rec["pmDEC"],
                        frame="icrs")
```