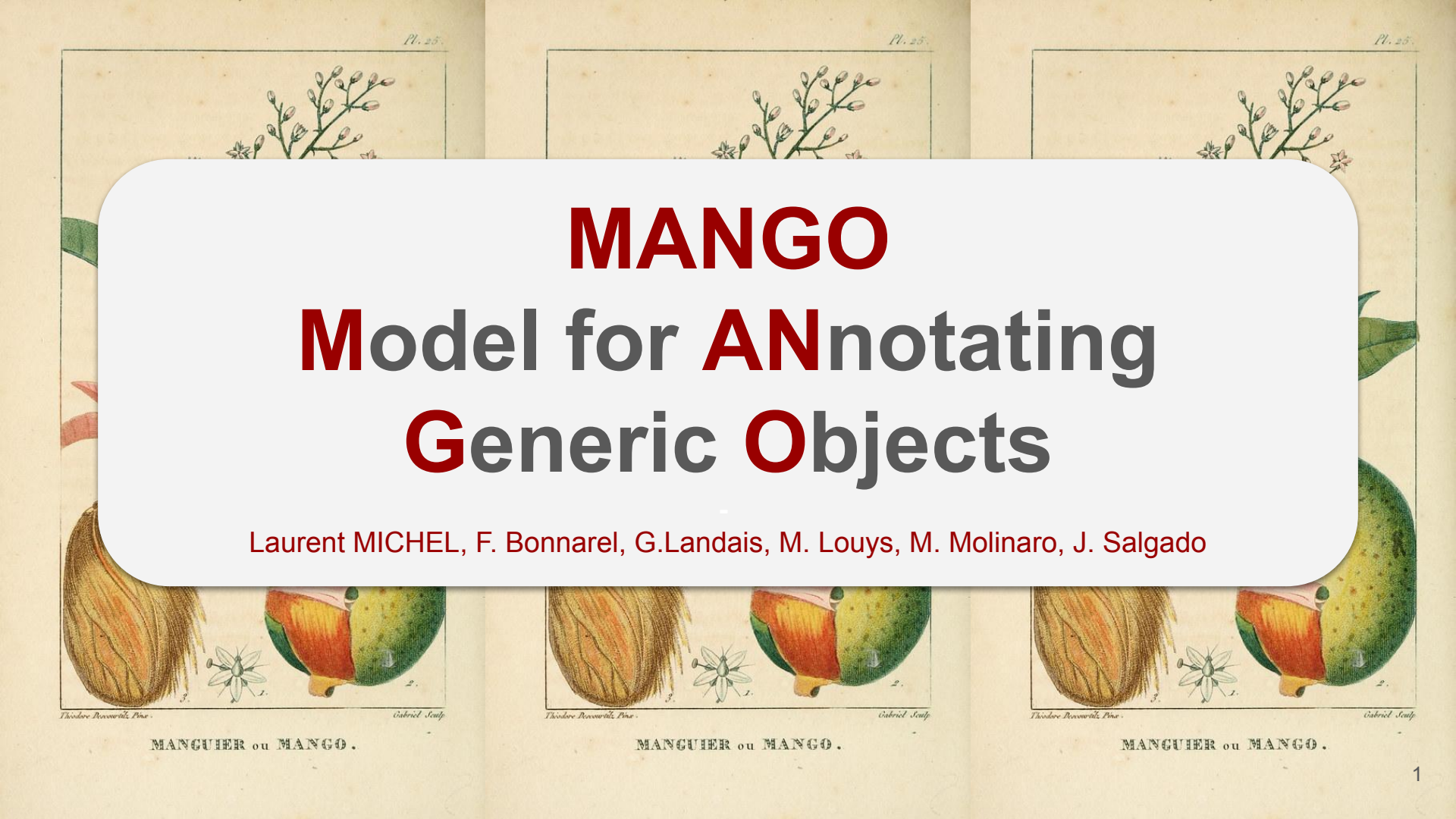


MANGO
Model for ANnotating Generic Objects

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

What MANGO Is for

Provide a **model view** on **tabular data**

- On data row ⇨ one Mango instance

Reconstruct **complex quantities** whose components are distributed at **different VOTable locations**

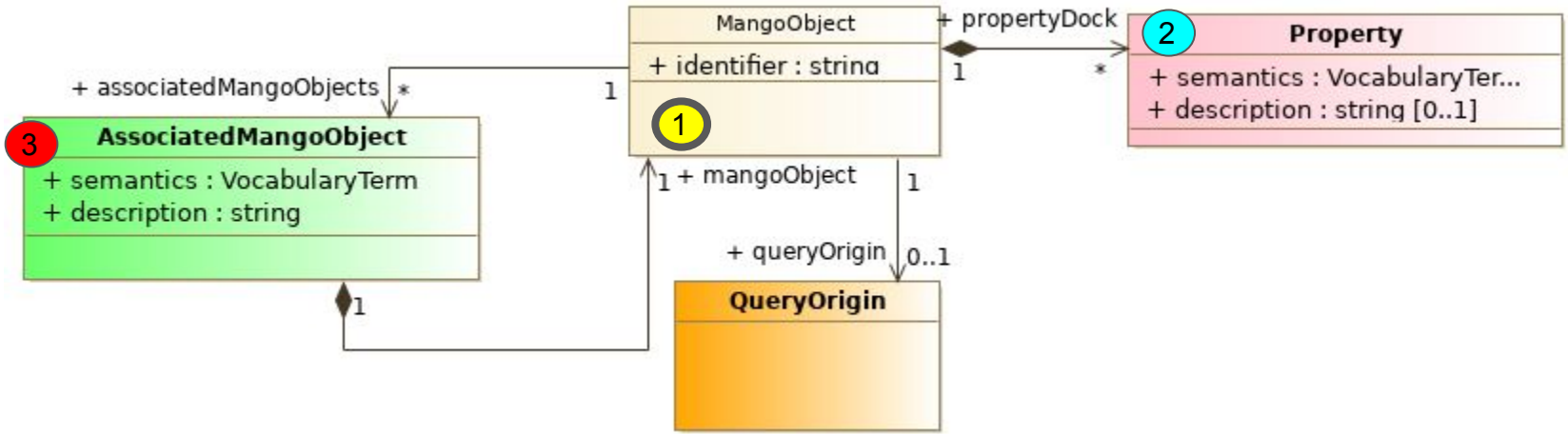
- `INFO/PARAMS/FIELD/COOSYS/TIMEYS`
- Missing meta-data (e.g. vocabulary)

- 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

What Are Complex Quantities (or Properties)

Quantities with **more** than **one coordinate**	• Position • proper motion • CCD position • errors
Quantity with **errors**	• many
Quantity with **specific coordinate systems**	• Photometric calibration/filter • Space/time coordinate system • Sparse axis
Quantity **linked** with **other quantities**	• Photometry + time stamps • Position + quality flag
Mix of all above cases	

Model Overview



3

Mango object association

- Sources with detections

1

Model Core

- Identifier
- Data origin

2

Dock hosting the properties

- EpochPosition
- Brightness
- Color
- Physical measurements
- Flags
- …

Another Model Overview

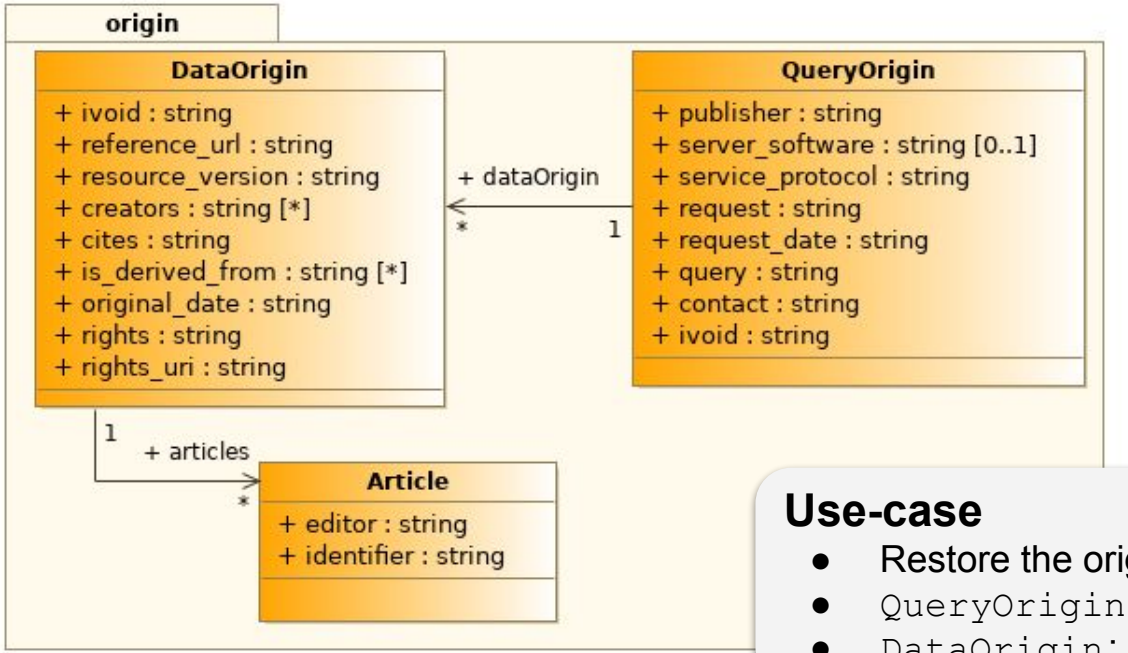
Market stall: Properties

Basket: MangoObject

Properties can be viewed as **independent models** and used as such….

… or they can be carefully embedded into a **MangoObject**.

Origin of the Mapped Data



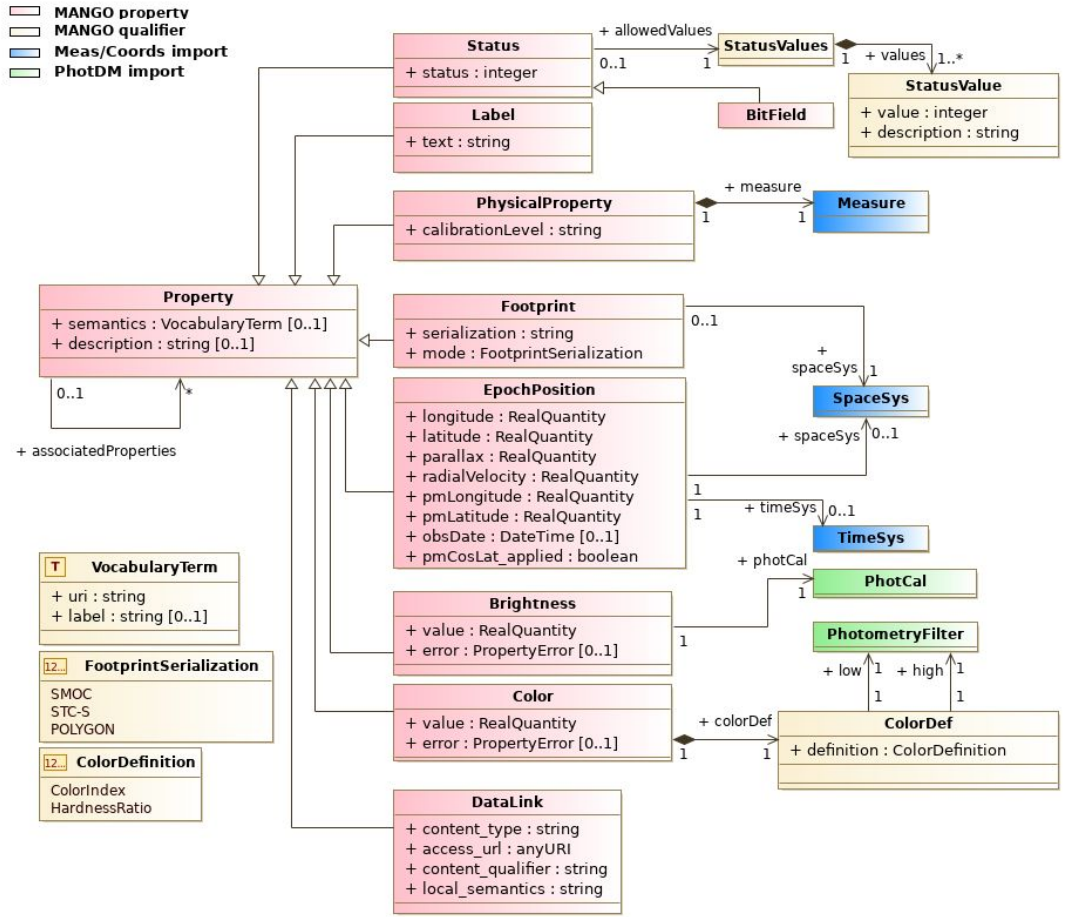
Use-case

- Restore the origin of mapped data
- `QueryOrigin` : Query that produced the VOTable
- `DataOrigin`: Origins of the queried data

Added value

- Structured view on **<INFO>**
- Allows to attach multiple data origin with one query
  - TAP join

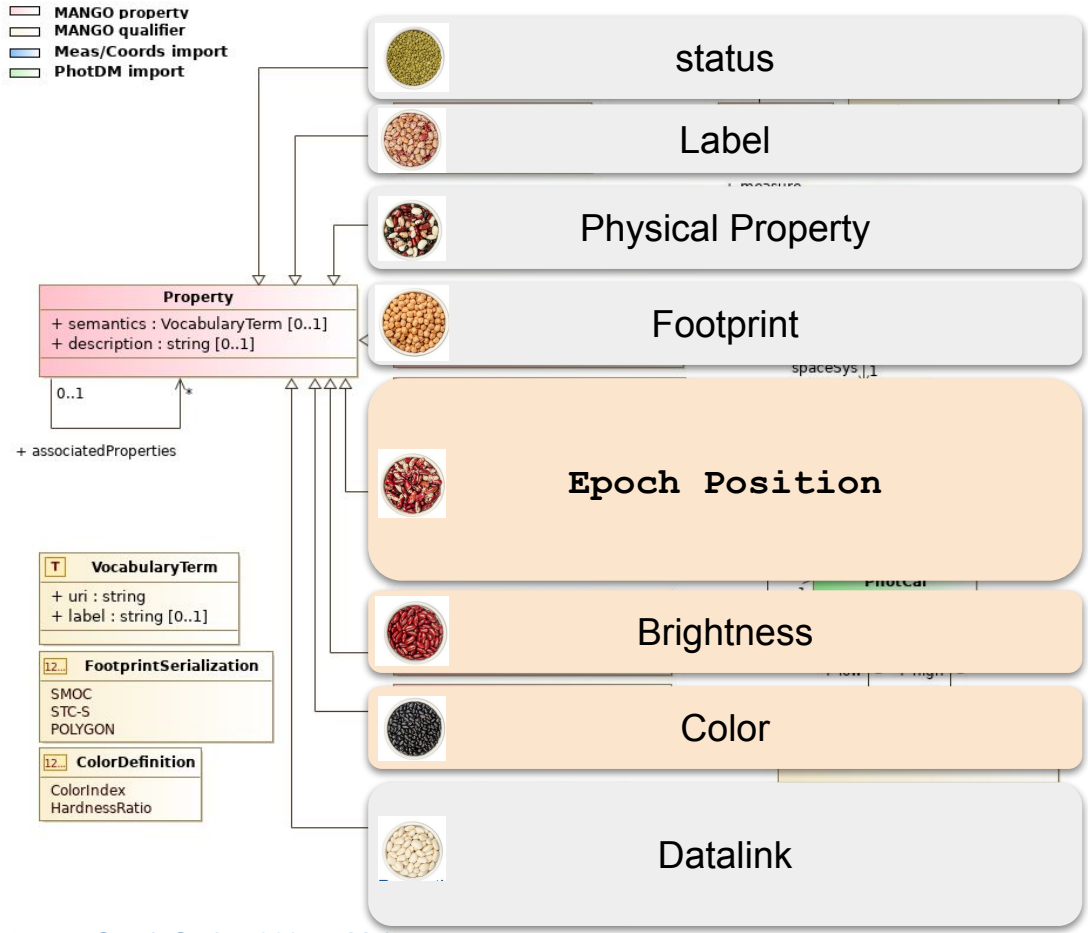
Supported Properties



A property is a **class modeling a quantity** that is present in one or more columns of the data table **extended** with a **semantics block**

- Property **values** can be set with **table data** or **literal** values
- Most of the properties are modeled with MANGO **built-in classes**
- Some can include **imported classes** (Coords or photDM)

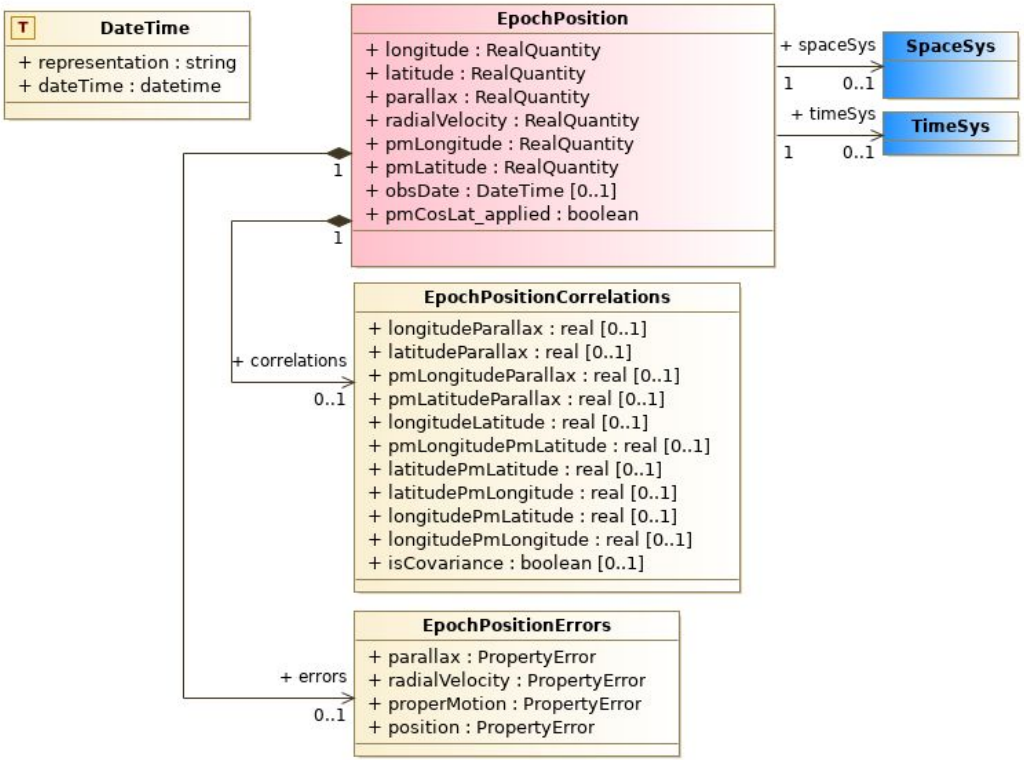
Supported Properties



A property is a **class modeling a quantity** that is present in one or more columns of the data table **extended** with a **semantics block**

- Most of the properties are modeled with MANGO **built-in classes**
- Some can include **imported classes** (Coords or photDM)

The Epoch Position Property



Added value

- MANGO flagship

All components in one flat class

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

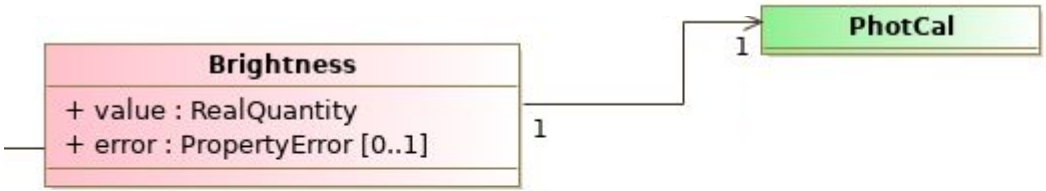
Correlation support

- Flatten correlation matrix
- A 6x6 matrix would be out of control

Implementations

- Vizier cone-search
- Xtapdb (see apps talk)
- Pyvo MIVOT package

Brightness Property



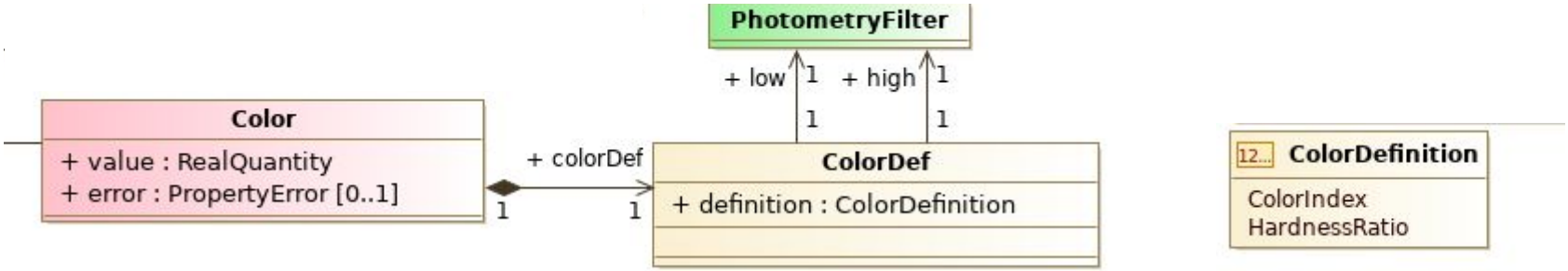
Use-case

- A model for the photometric data with error and calibration

Added value:

- Defines a place holder for photometric calibrations in VOtables
- Bind a magnitude or count rate value with
  - An error
  - A photometric calibration (imported from **`PhotDM`**)

Color Property



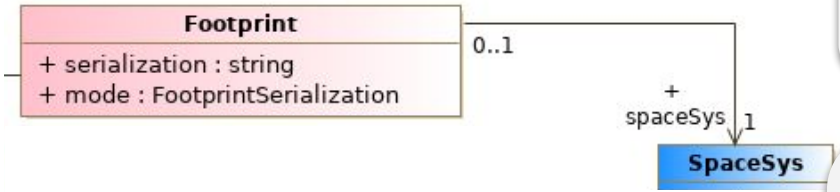
Use-case

- A model for the color data with error and filters
- Distinction between color index and hardness ratio

Added value:

- Defines a place holder for photometric filters in VOtables
- Bind a color or a hardness ratio with
  - An error
  - 2 photometric filters (high and low imported from **`PhotDM`**)

Footprint Property



Use case

- Standard description of extended objects
  - Gaz clouds

String serialization of complex shapes

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

Added Value

- Add a semantics to shape serialization

Support of Classes of the `Measurement` Model

Same pattern for all classes

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

Placeholder for classes of the `Measurements` model

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

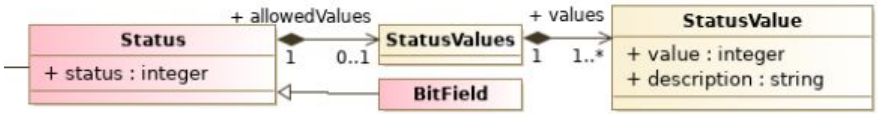
Great flexibility

- Accurate description of all axis
- Polymorphism allowing multiple representations quantities

Add a calibration level

- borrowed from Obscore
- http://www.ivoa.net/rdf/processing-level

Status Property



Use case

- Flag values
  - Detection quality
  - Source variability

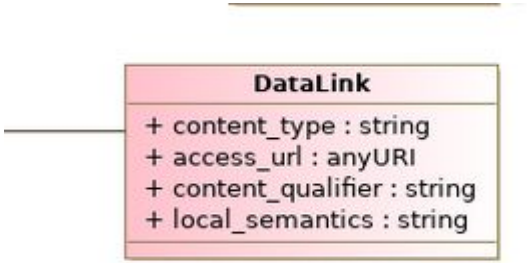
Set of allowed values

- Allowed values come with their descriptions

Added Value

- Make status values machine readable
  - Query setup
  - Value understandable by the client
- Give semantics to the status

DataLink Property



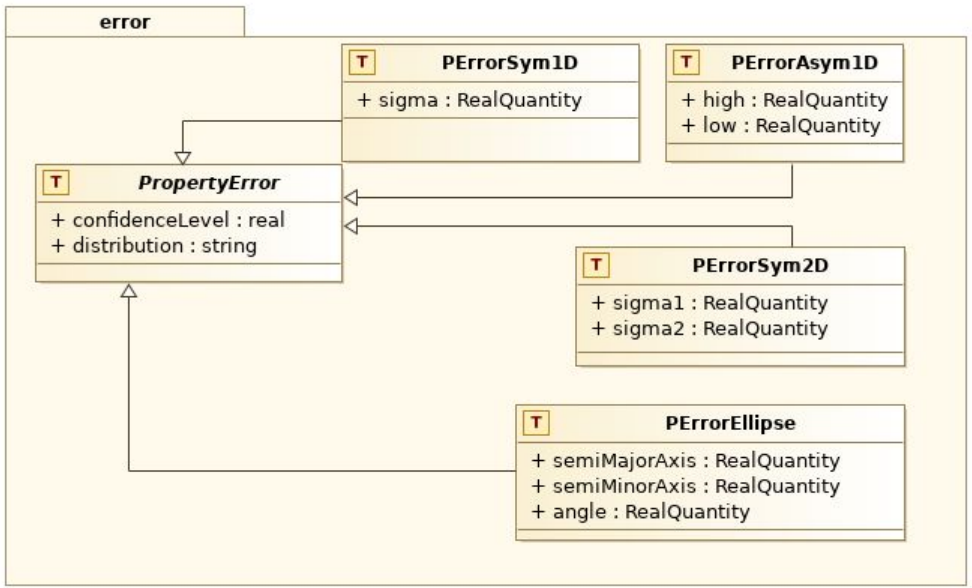
User Case

- Flat DataLink serialization for services exposing URLs in data tables and not running data link services

Added Value

- Add a semantics to flat URLs

Error Package



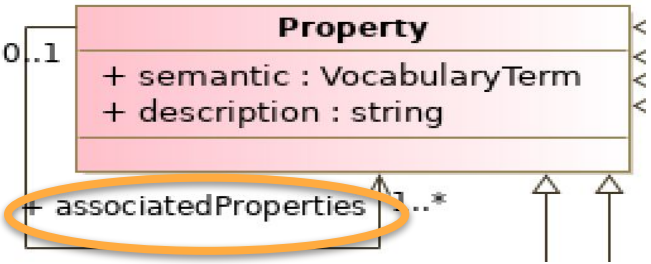
Use-case:

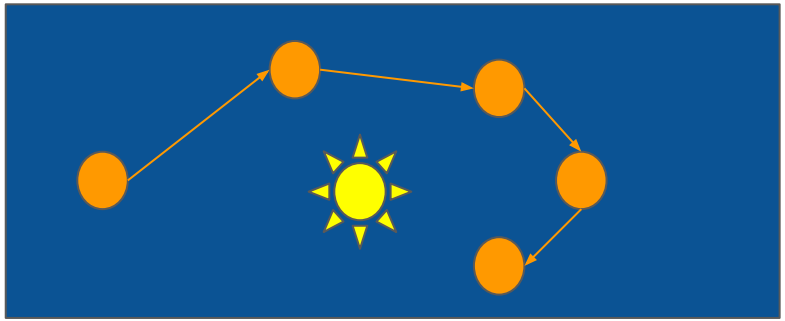
- Adapted to the EpochPosition
- Adapted to the cross-match

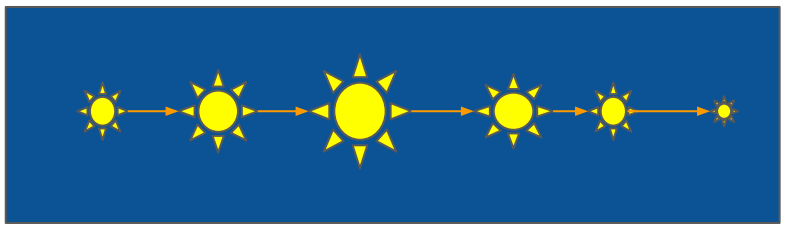
Added Value:

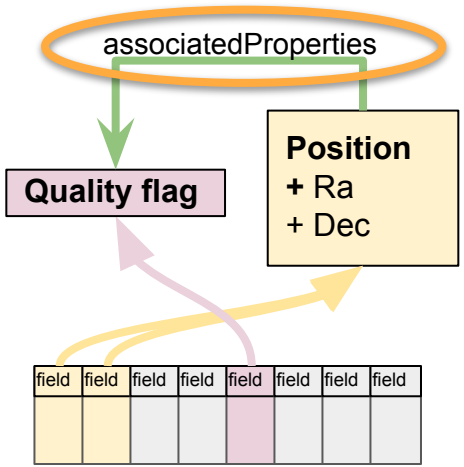
- Confidence level
- Statistical distribution
  - Both important for X-Match

Property Association









make it easier to interpret time-domain data.

- Variable objects

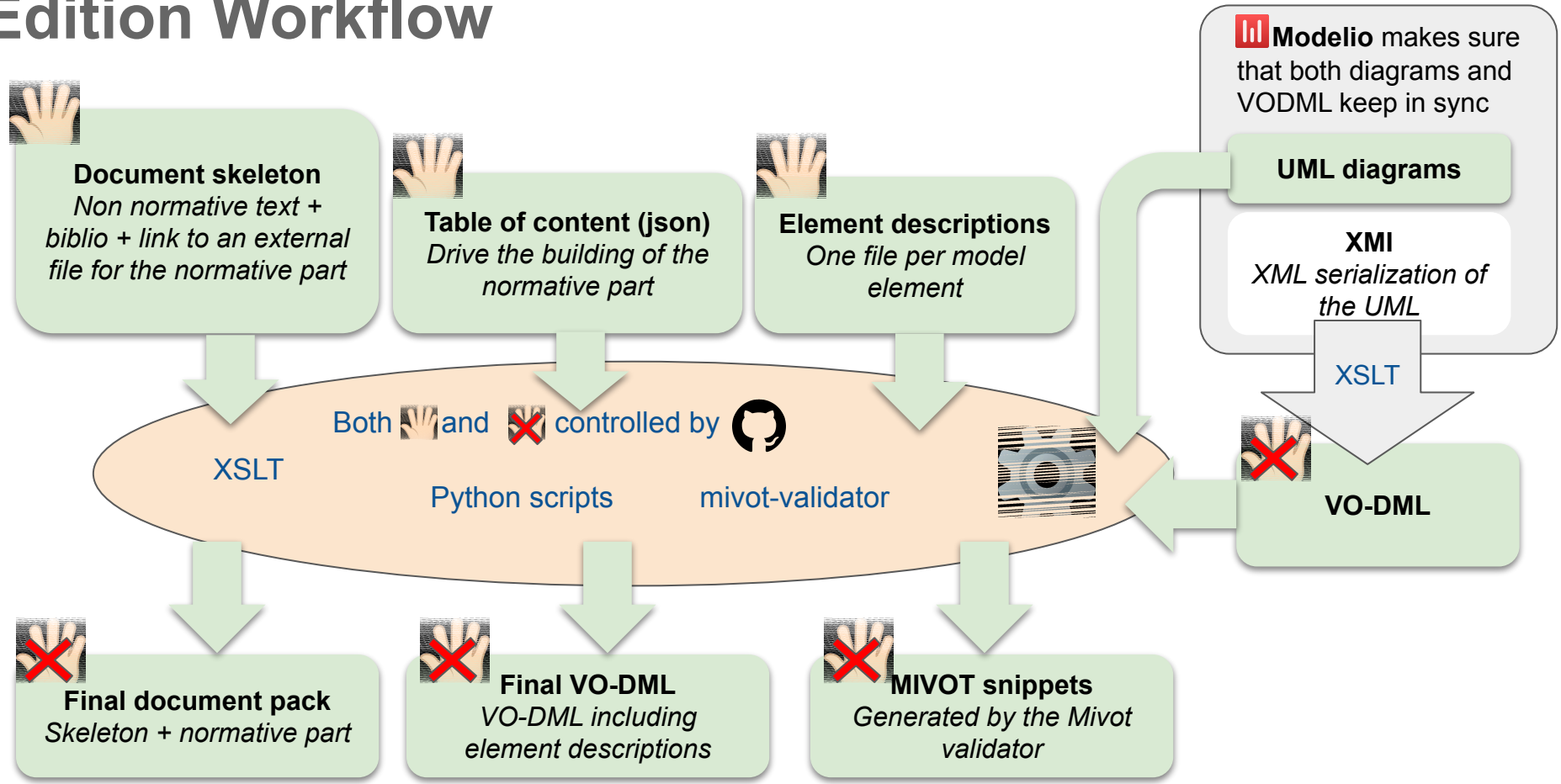
Composite objects

- Orbiting system

Associating properties with a flag makes it easier to filter data.

- Quality flag

Edition Workflow



Reference Implementations

4 static files

- Cover most of the features
- VOTable data mapped on MANGO with MIVOT
- Annotation possibly completed by hand
- Part of the project repository

3 services

- Vizier ASU (EpochPosition)
- Vizier ConeSearch (EpochPosition)
- XTapDB (XMM data, based on VOLTT, many properties)

PYVO API (1.7+)

- Consume VOTable data through the MANGO mapping
- Read EpochPositions as SkyCoord instances
- Helper to map data on MANGO
- More in apps session

Class	xtapdb	Gaia	datalink	vizier_cs_I_239
MangoObject	X	X	X	
QueryOrigin		X		
EpochPosition	X	X		X
Brightness	X	X		
Color	X	X		
FootPrint			X	
Status	X			
Label			X	
BitField				
PhysicalProperty			X	
DataLink			X	
PhotCal	X	X		
PhotFilter	X	X		
SpaceSys	X	X		X
TimeSys	X	X		
PErrorSym1D	X	X		
PErrorAsym1D		X		
APErrorSym2D		X		
Ellipse				
AssociatedProperties	X			

Status

A lot of work put in this standard.

- Document
- Services deploiement
- PyVO implementation
- Discussion
- Reviews

Ready for RFC

- Waiting for the IVOA green light

- See you on the RFC page.
- Meanwhile
  - https://github.com/ivoa-std/MANGO

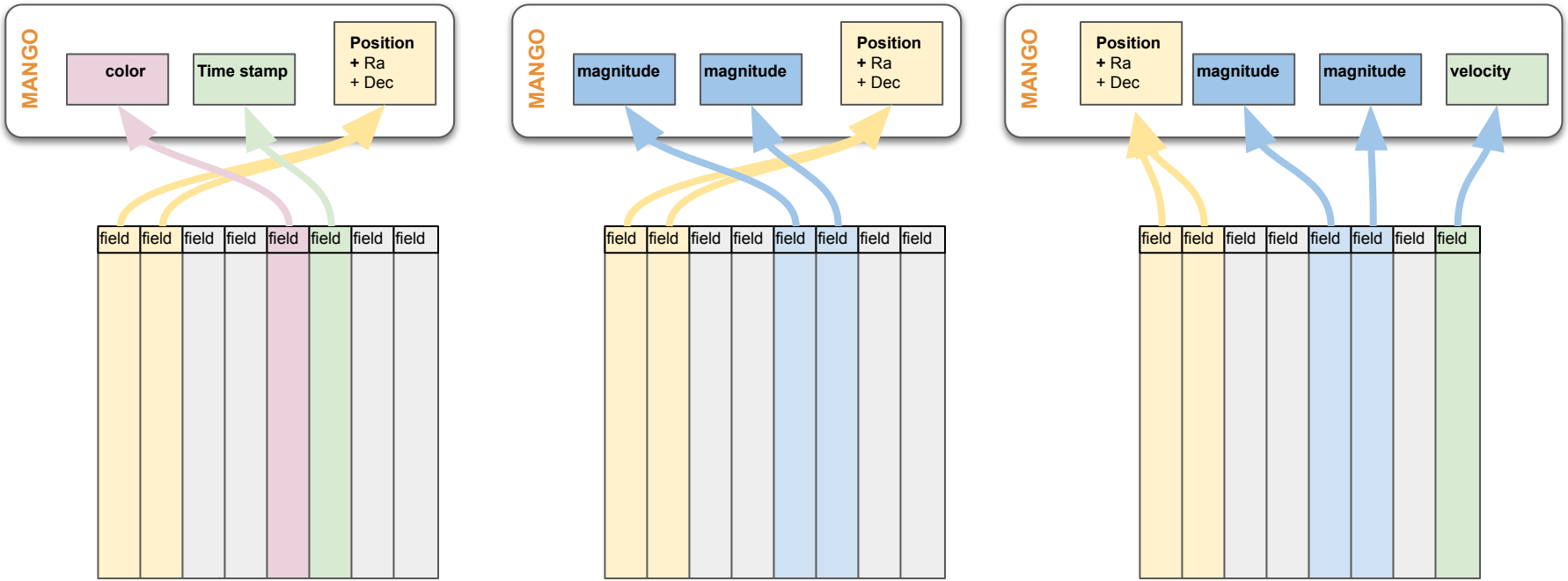
We will roll out the **red carpet** for anyone interested in **implementing** this framework

backup

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 meta-date 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)

Different Datasets - one Model



The model describes

- Global metadata
- Open set of properties
- Properties associations

The model does **not specify** any pattern of **expected properties.**