International
Virtual
Observatory
Alliance

Workshop (pre)Conclusions

Version 1.0

IVOA Note 2021-06-3

Working group
Data Model
This version
https://www.ivoa.net/documents/report-note/2021063
Latest version
https://www.ivoa.net/documents/report-note
Previous versions
This is the first public release
Author(s)
Laurent Michel
Editor(s)
Laurent Michel

Abstract

This note summarises the work done on the use-cases and give the preliminary conclusions that must lead a common strategy for improving interoperability with VO models.

Status of this document

This is an IVOA Note expressing suggestions from and opinions of the authors. It is intended to share best practices, possible approaches, or other perspectives on interoperability with the Virtual Observatory. It should not be referenced or otherwise interpreted as a standard specification.

A list of current IVOA Recommendations and other technical documents can be found at https://www.ivoa.net/documents/.

Contents

Acknowledgments

We thank all contributors who implemented the use cases or participated in discussions. We also thank the speakers who presented their views as developers or data providers.

Conformance-related definitions

The *Virtual Observatory (VO)* is a general term for a collection of federated resources that can be used to conduct astronomical research, education, and outreach. The International Virtual Observatory Alliance (IVOA) is a global collaboration of separately funded projects to develop standards and infrastructure that enable VO applications.

1 Introduction

The data model workshop has been initiated by the TCG in November 2020. It is a 4 steps process that must issue a global strategy for the model usage

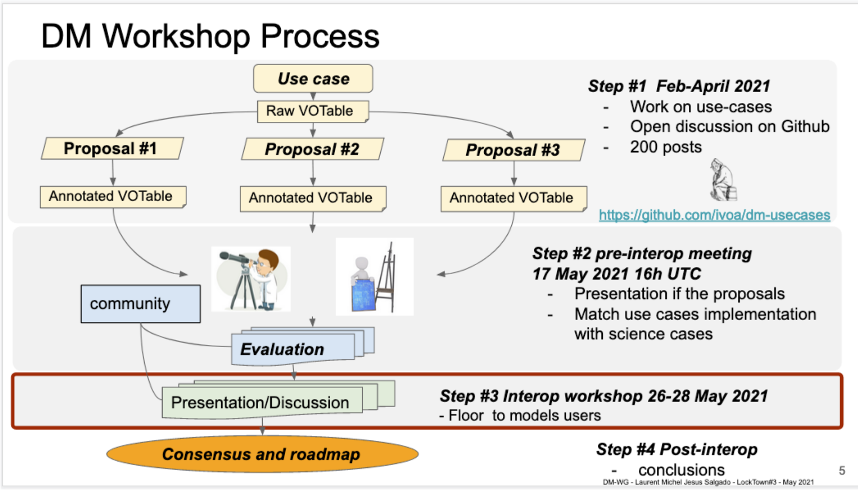

Figure 1: DM workshop

in the VO; this means sharing a common way to get a better interoperability by mapping data on VO models .

The primary goal of this workshop was to answer the following questions

- Do the proposed mode cover the science use-cases?
- What is the best mapping strategy?
- What is the most suitable annotation syntax.?

If we get a positive answer to all of the 3 questions, we will be able to propose a strait way to annotate data. Any missing information can be retrieved in the Github repository.

2 Tools and Models

2.1 Models

- **Coords (PR)** describes values associated with their coordinate systems.
- **Meas (PR)** describes measures as Coord instances associated with errors.
- **PhotDM (REC)** describes photometry filters, photometric systems, magnitude systems, zero points etc.
- **DataSet (WD)** Describes the structure and content of generic Dataset metadata for the IVOA.

- **CubeDM (WD)** presents an abstracted representation of NDimensional cube datasets.
- **Mango (WD)** proposes a flexible way to expose data related to astronomical source objects in an interoperable way.

In addition to this 2 ad-hoc models have been used:

- Model based on MANGO components for the complex time series (F. Bonnarel et al.).
- Model classes referring to Meas/Coord elements but with different structures (M. Demleitner).

2.2 Mapping Syntax

Two different syntaxes have been used to annotate the use-cases datasets.

- **VODML Mapping (WD)** proposed by G. Lemson et al. in 2018.
- **ModelInstanceInVot (WD)** proposed by L. michel et al. in 2020.

It is to be noted that ModelInstanceInVot derives from VODML Mapping.

2.3 Clients APIs

- **Astropy extension (Markus Demleitner)** designed to process the annotation as proposed by Markus Demleitner.
- **Rama (Mark. C. Dittmar et al.)** Python API based on generated code designed to process VODML Mapping annotations in connection with Astropy.
- **modelinstanceinvot-code (L. Michel et al.)** Python API based on dictionaries designed to process ModelInstanceInVot annotations in connection with PyVO.
- **AWK scripts (Mireille Louys)** shell script showing that ModelInstanceInVot annotations can be read with basic shell scripts.

3 Use Case Implementation

3.1 Use Cases

5 use-cases on 10 have been selected to assess te conclusions. This choice covers most of the current science cases. Table 1 gives a brief description of them. All details can be found on Github.

Case	Purpose
time series	identifying the whole thing as a time series, Identifying the independent axis/axes, associating values and errors Figuring out the observation position This use-case comes with one simple time-series and 2 more complex one (Gaia ZTF) where photometric points are mixed in data tables
standard properties	Allow clients to easily retrieve scientifically relevant source properties
combined data	Aggregation of various pieces of information into a single VOTable
precise astrometry	Client authors want to recognise all information in catalogues relevant to precision astrometry (e.g., distances or radial velocities as necessary in foreshortening calculations)
column grouping	The data table must be annotated in a way that a client can easily detect column groups. These groups have no particular semantic. They are just telling that the measures are related each to other.

Table 1: Use-cases selection

3.2 Implementation

Matrix 2 gives an implementation overview. All details can be found in the Github repository.

Case	MD.AS	C.VM	C.MIV	FB.MIV	M.VM	M.MIV	V.OF
simple time series	X	X	X				
Gaia time series	X		X	X			
ZTF time series	X		X	X			
std properties					X	X	X
combined data					X	X	
precise astrometry	X					X	
column grouping					X	X	X

Table 2: Implementation matrix (MD: M. Demleitner, C: CubeDM, FB: F.Bonnarel, M: Mango, AS: annotation scheme, VM: VODML mapping, MIV: ModelInstanceInVot, V: Vizier, OF: On the fly)

4 Requirements

The main requirement shared by all the stakeholders is not to break existing things but to add missing features. There is no strong demand for working with data mapped on models, but there is a real interest to use this mechanism to fill some gaps and to implement new functions.

4.1 Science Requirements

- VOTable annotations APIs must be included in AstroPy or PyVo,
- Need for making multi-body data (asteroids, planets...) interoperable
- Need for standardising multi-messenger meta-data
- Need for model for photon lists
- Need for an accurate time model for moving objects
- Need for a model for orbital information
- Need for a better support for X-ray data (photon based datasets, source-detection or spectrum-model association, probabilistic errors)

4.2 Data Provider Requirements

- Need for adding data provenance
- Need for grouping measures
- Need for a better description of the photometric calibrations
- Need for a data annotation validator
- Need simple views on complex models
- On the fly annotation requires stable building blocks

4.3 Client Developer Requirements

- Data annotation must not come in replacement of existing processing but it must add features that are currently missing such as a better coordinate systems or axis descriptions.
- Need for a full spectrum characterisation.
- VOTable annotations must come with reusable libraries at least in Java/Python

5 Conclusions

It has been decided before Interop to withdraw the approach consisting of mapping data on a set on independent model components and hence to keep working with integrated models re-using Meas/Coordinates/PhotDM classes.

5.1 What Worked Very Well

- Data contained in all 5 use-cases can be modelled with existing models
- The 2 annotation syntaxes have been used with success
- All APIs have been able to process all annotations.
- It has been demonstrated that Enabling AstroPy or PyVo to work with annotated data is feasible without breaking the regular data access.

5.2 What Has to Be Improved

- The connection of PhotDM with Meas/Coordinate/Mango/Cube has to be tuned. This is a pending issue while the PhotDM VODML/XML has not be released.
- Some update of Meas/Coordinate have been suggested, especially for the sky positions.
- A new class hierarchy has been proposed for the MANGO Parameters.

5.3 What Has to Be Discussed

- It is very desirable to finally have one syntax. There are ongoing discussions to merge both proposals
- Extending our models to cover new use cases,