How to use Data Models to Enhance **Interoperability** in the VO

Workshop Wrap-up

Chicken-Egg DM Deadlock

Let's consider we have working proposals for both models and annotation

Client developer
I'll support model mapping **when data providers** will annotate their data

Data provider
I'll annotate my data when **client developers** will be able to process annotated data

Final users won't break the loop
I can live without annotated data

All Have Good Arguments

Client developer

What could I do with the models that I can't do now?

Dataset will never be all annotated; I'll ever have to process *raw* dataset

What happens when a model is updated?

Data provider

Annotating data is a big job for my service, what is the gain?

What happens when a model is updated?

All Have Good Arguments

What could I do with the models that I can't do now?

- Get a clean representation of the coordinate systems
- Get a clean representation of the errors
- Support cross-columns parameters
  - Columns grouping
  - Complex errors (pos + pm + parallax)
- Gather data from multi-table VOTable (sources + detection)
- Exchange model instance with SAMP

We need models and tools to convince people to go into that way
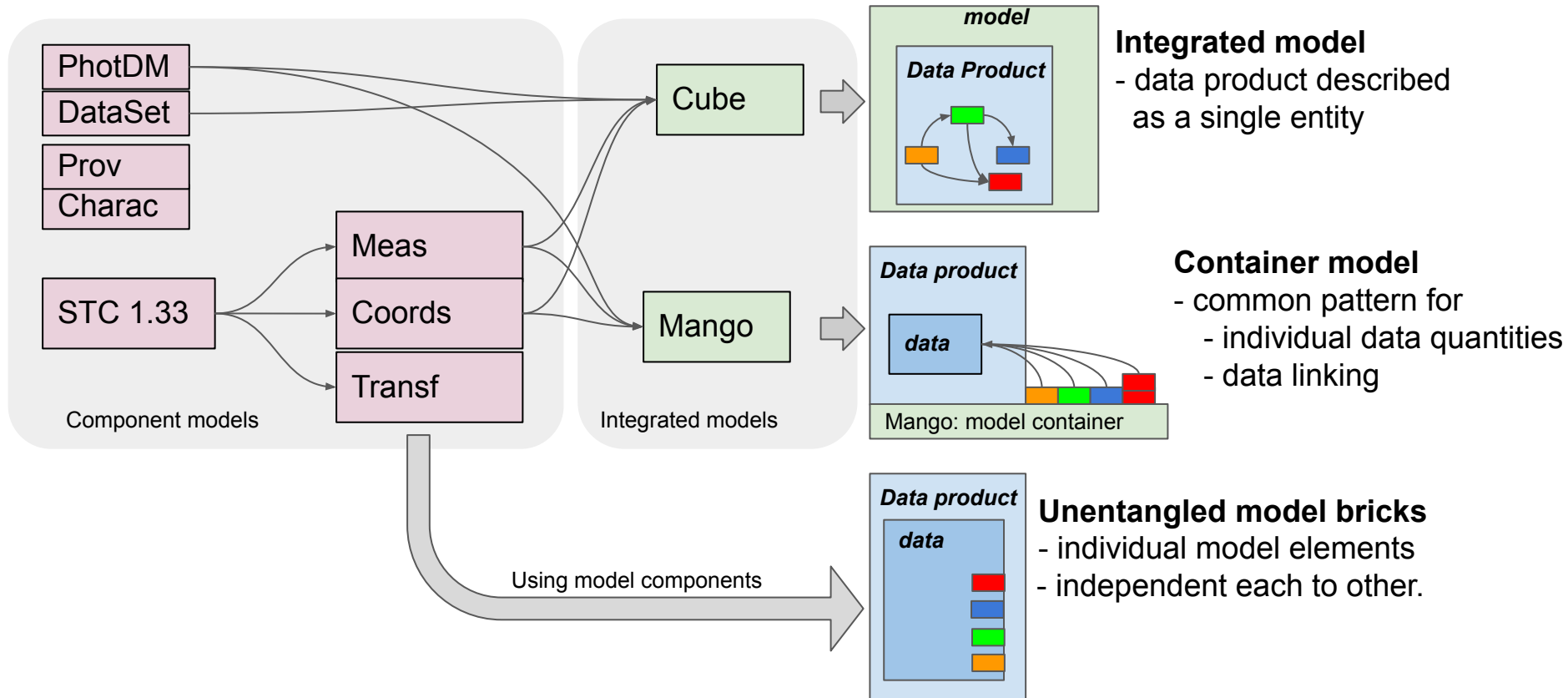**But we don't have**

The DM workshop

- **Mapping data on VO models**
  - Provide clients with enough material to build model instance from the data.
  - We need consensus on
    - the models to be used
    - the annotation syntax

a bit stuck so far

Reason #1: Different Approaches to Annotate Data

Integrated model
- data product described
  as a single entity

Container model
- common pattern for
  - individual data quantities
  - data linking

Unentangled model bricks
- individual model elements
- independent each to other.

Reason #2: Different syntax proposals

VO-DML Mapping (Gerard Lenson, Mark Cresitello Dittmar)
- Original proposal (2018)

Works for all use-cases + client code

ModelInstanceInVot (Laurent Michel et al.)
- Proposal for a more condensed syntax

Works for all use-cases + client code

DM Workshop

There are 3 top level challenges:

1. Showing whether MCT/PhotDM cover our needs
2. Adopting a mapping strategy
3. Adopting a mapping syntax

The challenge of the workshop is to trace a path leading to a community agreement.
The goal of this pre-interop meeting is to give to attendees (who are more science than VO bargain aware) a clear view on the problem as well as on the different proposals.
If we do it properly, we can expect to get some people involved enough to contribute to the workshop.
For this, we have to operate the meeting in way that facilitates the understanding of the different facets of the issues and especially to get the pro/con of each proposal.

DM Workshop

DM Working group committed by the TCG to run a workshop to build a consensus that can get the **back on the road**

FEBRUARY 2021: use case workbench setup
- Use-case: raw data
- Proposal: Annotated VOTable + documentation
- Issues + Wiki

MARCH-APRIL 2021: open contributions
- Exercice
- Comment
- Proposals

MAY 2021: Report at Interop

July 2021: Wrap-up

Use Cases

- ## Mostly based on real data
  - Vizier Chandra Gaia Gavo ZTF

- ## A wide range of data
  - Time series,
  - Standard quantities: positions, magnitude, velocity
  - HE data: flux, hardness ratio, flags
  - Data grouping: multiple frames, grouped columns, complex errors
  - Combined data (multi-table)

Clear-cuts Positions on Models

Un-entangled models (Markus Demleitner)
- A mapping block on the top of the VOTable
- Contains sparse model components not gathered in a global model

Works fine for the simple case

Product models (Mark Cresitello Dittmar)
- A mapping block on the top of the VOTable
- Maps the data on a product model (e.g. Time Series)
- The product model is made of model component classes

Common interface for any products of a given type

In between (Laurent Michel et al.)
- A mapping block on the top of the VOTable
- Contains sparse model components arranged in a container (MANGO)
- Support both parameters and associated data

Suited for archival data even with complex features

Clear-cuts Positions on Models

- **Un-entangled or integrated models?**
  - 2 months of discussions
  - 200 posts
  - No agreement or convergence either
- **May 2021**
  - TCG made the decision of **working with integrated models**
    - Better suited for foreseen more complex data
    - Preserving the modeling efforts already achieved

Key Facts Workshop after Interop

- Models:
    - MC, Cube and MANGO accepted
    - New use-cases
        - X-RAY Astronomy
        - Asteroids, multi-core datasets
        - CTA and MM astronomy : meta-data characterization

- Data Provider / client
    - Need for annotations to help processing spectra
    - Need to associate parameters
    - Need for a simple description of the photometric calibration
    - Need for a simple view on Provenance
    - Annotation on the fly feasible
    - Model-based PyVO API easy to design

- Mapping syntax
    - Divergence between the 2 proposals (VODML mapping and ModelInstanceInVot)
    - Proof of concept for a YAML serialization of model instances

Mapping Syntax: Merged Syntax on the rails

Roadmap

- Meas Coord
  - New Lon/Lat point class
  - Complete the RFC
- Cube
  - Start the PR process
- Mango
  - Implementation of the the modification suggested while the workshop
  - Start the PR process
- Mapping syntax
  - Write the spec
  - Provide Python parsers
  - Start the PR process

What are your expectations from these huge efforts?
- Have you some?
- Was it useful?

Model

Vocabulary

Not agreed

Annotation

UTypes

Neither do I

<GROUP>

VOTable
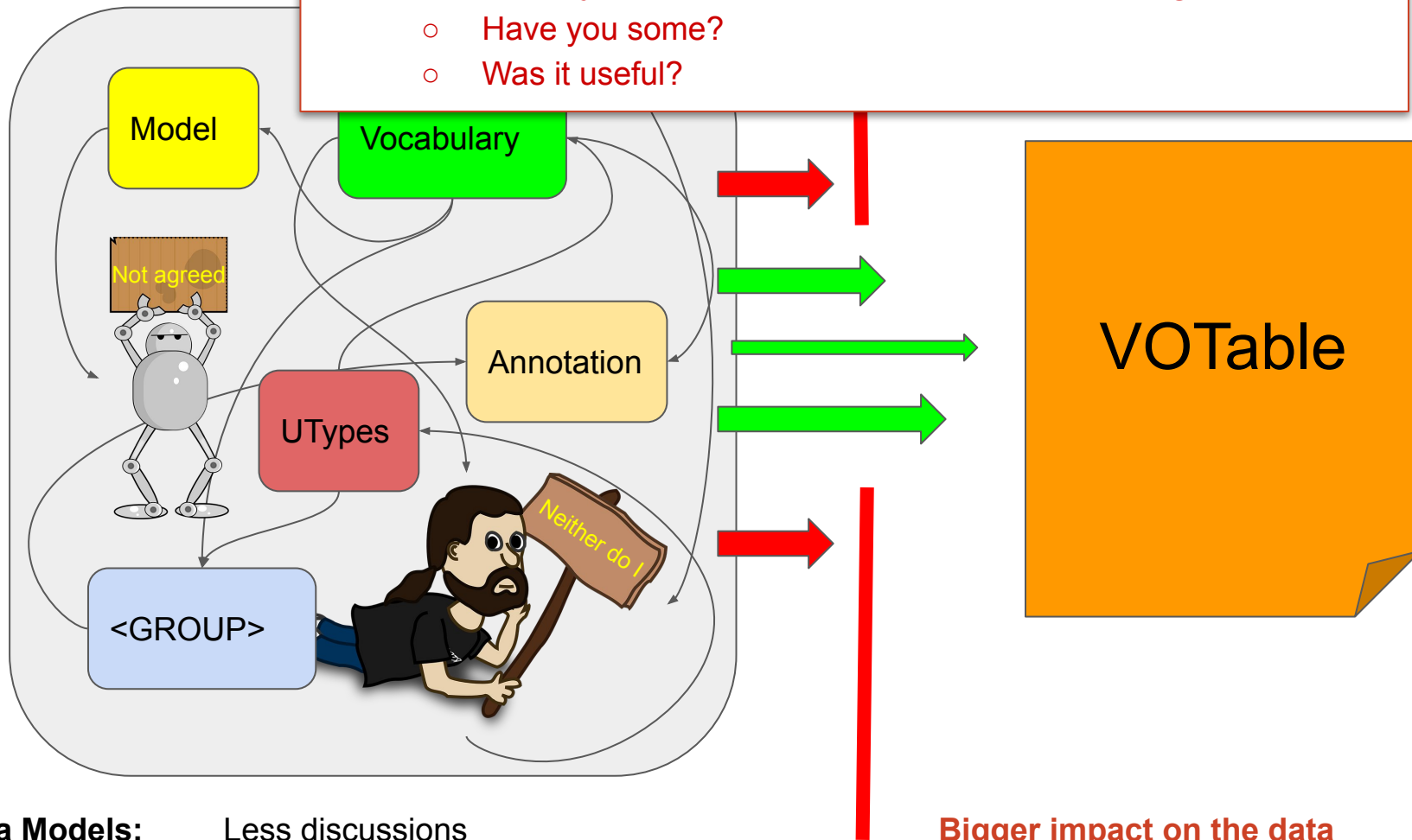
Data Models: Less discussions

Bigger impact on the data