# DM Implementation Requirements
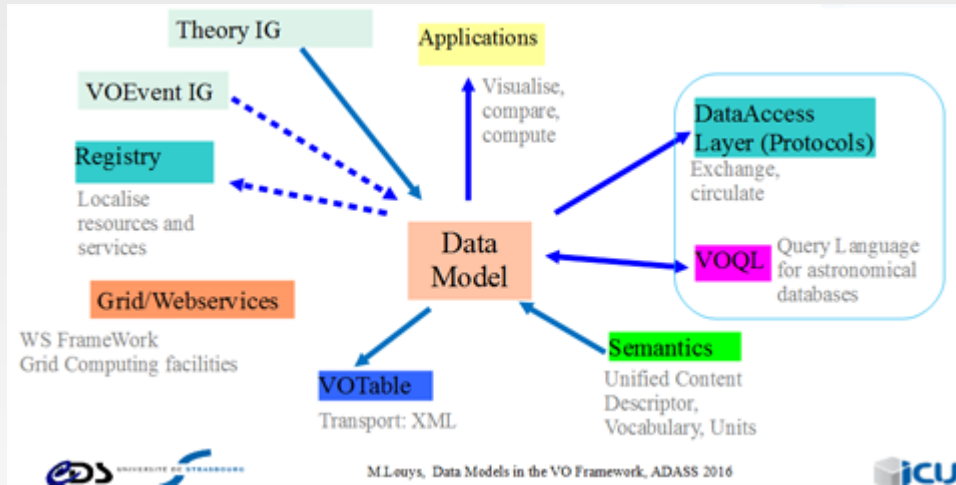
## Mark Cresitello-Dittmar, SAO

# Document Standards

+ General standard

  – demonstrate two interoperable implementations of each feature

  – validation tools should be available


* We want to keep in theme with these requirements

# DM standards



M.Louys, Data Models in the VO Framework, ADASS 2016

A data model is used BY access protocols, applications, and serialization standards to inform their design and facilitate interoperability.

A Science use case typically involves implementing several of these aspects.

To require a data model standard to have the entire science use case implemented is too dissociated from the model itself.

# Example

+ As an extreme (hypothetical) example:

- image cutout use case requires Matrix transform
- We decide to generate a Transform model.
- The Transform model gets a requirement to support Matrix transform, with details about size/shape etc.
- That model is written (about 15min later)
- working group consensus is reached

+ This model should be able to go forward without waiting for two institutions to implement:

- access protocol
- cutout service
- data serialization of cutout image
- visualization application consuming the image

# Risk

As use cases are implemented, they may discover missing and/or incorrect requirements

- – minimized if accompanied by real world examples

# Initial Proposal

Initial Proposal:

- – If described using VO-DML, model must be validated
- – Serialization (XML/VOTable) of instance, which can be validated
- – X Does not guarantee usability by software unless we have a vetted serialization standard.
- – X May not apply to 'real world' data

# Response themes

+ From email thread started in Stellenbosch

  – DM is designed "to meet requirements expressed in formalized use-cases, to be valid, it must meet these requirements"

  – Serialization should be validated.. constraints, datatypes and vocabulary

  – Serializations should work on 'real data'

  – In lieu of serialization standard, having an application consume instances independently serialized according to some agreed form.

  – Without serialization standard, cannot have automated validation, so must 'work around' it..

  – generate serialization which is validated 'by hand' to show that it represents the model content.

  – Serialization alone will not guarantee usability, should be a service or application using serialized instance

# Proposal

We have to ensure that the model:

- is valid (eg: vo-dml validation)
- covers its requirements
- is serializable (can define instances)
- is usable by software (clients)
- * by someone other than the modeler(s)

By providing:

- serialization that touches each entity of the model, possibly fake
- real world serialization(s) covering the use cases the model is designed to fulfill
- client software which can read the serialization and demonstrate it can 'understand' the contents

# Discussion

Discussion:

- Agreement on proposal?
- 'sample' serializations by modeler? (~unit test)
- real world serializations by external participants?
- what format?
    - wait for serialization standard (vo-dml:Mapping)?
    - XML
- what client software?
    - 'dmlist' utility?
    - validator?
- Apps group involvement?