International
Virtual
Observatory
Alliance

A Component and Association Based Model for Source Data Version 0.1

IVOA Note 2019-09-22

Working group
Data Model Working Group
This version
http://www.ivoa.net/documents/cab-msd/20190922
Latest version
http://www.ivoa.net/documents/cab-msd
Previous versions
This is the first public release
Author(s)
François Bonnarel, Tom Donaldson, Laurent Michel, Marco Molinaro
Editor(s)
Laurent Michel

Abstract

This note proposes a possible way to make source data interoperable. It takes into account the huge diversity of source data in term of both format and usage. Both data annotation and annotated data parsing processes are also considered in this document.

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 http://www.ivoa.net/documents/.

Contents

Acknowledgments

Paris session speakers TDIG members

Conformance-related definitions

The words "MUST", "SHALL", "SHOULD", "MAY", "RECOMMENDED", and "OPTIONAL" (in upper or lower case) used in this document are to be interpreted as described in IETF standard RFC2119 (Bradner, 1997).

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 source DM is a long term concern for the DM working group and more generally for the IVOA. In the past years, there were some proposals to design a global model for sources (Salgado and Lemson et al., 2016) of for catalogs (Osuna et al., 2006). Other proposals, more model-agnostic, were focused on the data annotation in VOTables (Demleitner and Ochsenbein et al., 2016) (Derriere, 2016). In this case the goal was no longer to design a source model but to provide a complete description of individual quantities (positions, velocity. . .). None of these proposals succeeded for reasons that are not discussed here.

The source DM issue resurfaced at the spring 2018 Interop in Victoria during an hands-on session focused on the tools available to work with VO data models and especially with VO-DML. The goal of this session was to annotate data from different origins in order to make them interoperable with each other. The main concern expressed during this session was not related to the tools themselves but to the lack of models for sources. This is a big paradox in the VO world ; source data which represent the basic bricks of the astronomer work, have no model. This paradox can be explained by the fact that sources data are multifaceted. The way of which source data are organized depends on the survey they come from, one the way they have been generated and on the expected use. In a more general way, it depends on the science we want to do with them. This diversity cannot be endorsed by a single model. Having a global source model would lead to a very complex solution not usable in practice.

1.1 A Model in Between Component Models and Product Models

IVOA models can be split in 2 classes. The component models, usable in various context for various data products (STC, Characterization . . .) and the product models (e.g. NDCube, Spectrum DM) describing each one specific science product. Source data do not match these 2 categories. They need component models to describe individual quantities but they cannot fit within one single product model just because there is no science product type enclosing all possible usages.

1.2 The Paris Session

In early 2019, we proposed to resume the source DM project from input given by different people and taking into account the new VO landscape (VO-DML, STC, Astropy, TAP,...). In spring 2019, DAL-WG, Apps-WG and DM-WG have arranged a joined session in Paris where people involved in new surveys, data curators and client developers were invited to present their requirements. The present note makes a coarse synthesis of these requirements and outlines a proposal capable of ensuring the interoperability between source related data. The framework is referenced as CAB-MSD (Component and Association Based Model for Source Data - until better:contest open) in the note.

2 Outcomes of the Paris Session

The talks given in the Paris session addressed 3 different points of view of the source data operations: 1) data providers point of view, 2) data curator point of view and 3) client developers point of view. The list of requirements issued from this session is still open but we can reasonably consider that for now, there is no incoming use case fundamentally different from those presented below. In addition we provided demands coming from the IVOA community for time series.

2.1 Data providers

The requirement of some data provider are summarised in table 1. More detail are available on the session agenda page [1]

2.2 Data Curators

Vizier (table 2) is a specific case; unlike the surveys mentioned above, it would have to apply CAB-MSD to a large variety of existing datasets daily updated. The easiness of data annotation is very critical for Vizier.

Although not mentioned in the session, TAP services represent a case similar to Vizier in a sense that datasets have to be annotated on the fly. In a perfect world, a TAP server should annotate the queried quantities by using tags set in the TAP schema (UType columns in the TAP_SCHEMA tables). This implies that the TAP schema is properly set. This feature remains on the edge of CAB-MSD but it should be kept in mind for the design of data annotation mechanism (see appendix A).

[1] https://wiki.ivoa.net/twiki/bin/view/IVOA/SourceDM

Survey/Project	Speaker	Required data content
Gaia	J. Salgado	identifier, reference position, proper motion, parallax + distance, correlation, source extension, radial velocity (redshift), luminosity, date, multiple detection
Euclid	J. Salgado	identifier, position, Reshift, correlation with Gaia, photometry (ground + sat), morphology, reshift, photometric redshift
Exoplanets	M. Molinaro	position, orbit, different source level (star, planet, moon), status and classification, orbiting system description
Morphologically Complex Structures	M. Molinaro	morphology
Chandra	F. Civano	detection (name, pos, time, extension, PHA) All quantities are time dependant, Dependant on calibration + physical model

Table 1: Data provider requirements

Service	Speaker	Required data content
Vizier	G. Landais	pre-existing data, grouping columns, lots of available metadata, column name formatting, filter service implemented, one column different frames

Table 2: Data curator requirements

2.3 Client Developers

In addition to standalone clients (table 3), more and more users are using programming language APIs to analyse data (e.g. Astropy). In this context the capability of describing individual quantities in query responses is very valuable. This would allow the user to easily extract the quantity he or she needs out of the scope of any domain-specific model.

2.4 Time Domain

The time domain requirements have been discussed in several TDIG sessions.

Tool/Service	Speaker	Required data content
Aladin	P. Fernique	position, time, flux, link, FoV, column grouping
X-match	P.X. Pineau	identifier, position, proper motion, photometry

Table 3: Client developer requirements

Interest Group	Speaker	Required data content
Time Domain	A. Nebot	identifier, position, associated products, photometry, Timestamp

Table 4: Time domain requirements

The Time Domain interest group (table 4) agreed that most of the current use cases are covered by a model defining time series as tables of timestamp/photometric points. However, the notion of time series can be understood in a wider scope. A time series is a multi-dimensional structure. The first dimension, the independent axis, is the time, and the others, the dependent axes can be anything. They can be magnitudes, velocities, positions, spectra, image or anything else. It is impossible to describe this with a regular model. Michel (2017) showed a possible solutions based on the usage of model references into the data mapping, but this approach has not been continued. This use case must however be part of the CAB-MSD requirements.

3 High level requirements

3.1 Model Requirements

The above list of requirements suggests that the source model must support 3 sorts of data:

- *[M1]* Support of any sort of numerical measurements with their coordinate frames and units
- *[M2]* Support of shape descriptors
- *[M3]* Support of associated data

The set of data describing a specific instance of source highly depends on the peculiar context.

- *[M4]* The model must keep unchanged whatever the scientific context is.

3.2 Data curator requirements

The following requirements have not been discussed in Paris but they result from discussions recurrent in the IVOA. The notion of annotation refers to the tags to be inserted into the data files to map data with the model.

- *[DC1]* The CAB-MSD mapping must be an add-on for working services:
  - *[DC10]* Original data must not be altered by CAB-MSD annotation
  - *[DC11]* CAB-MSD annotation must be implementable in existing services.
- *[DC2]* The data annotation must be limited to quantities with a real scientific interest. Annotating data has a great cost for the curator team, thus this action must be restricted as much as possible.
- *[DC3]* CAB-MSD must be able to map complex quantities.
  - *[DC31]* Quantities shared among multiple columns
  - *[DC32]* Quantities spread over multiple tables.
  - *[DC33]* Missing metadata can be added as literals
- *[DC4]* The CAB-MSD implementation must be designed in a way that facilitates the use of templates making easier the annotation process.
- *[DC5]* The CAB-MSD implementation must be designed in a way that quantities can be mapped in a independent way.

3.3 Client Requirements

Before to be discussed in Paris, these requirements have been discussed many times in IVOA .

- *[C1]* APIs processing datasets annotated with CAB-MSD must be capable to discover which quantities are present:
  - Does this VOTable support CAB-MSD?
  - Where is the main table?
  - What are the quantities contained in that table?
  - Is that specific quantity available in my dataset?
- *[C2]* CAB-MSD annotation must allow clients to correctly interpret any annotated quantity present in the dataset.

  - Which filter has been used to compute that magnitude?
  - What is the epoch of that position?

- *[C3]* CAB-MSD annotation must allow clients to access and to interpret linked data.
  - Are there linked (joined) tables?
  - Are there other detections for this source?
  - What does that URL return?

- *[C4]* The code must be independent of the searched quantities.
  - *[C41]* Extending the quantities supported by CAB-MSD must not require code update, at least for the data extraction. This requirement is not valid for data processing.

- *[C5]* APIs in different languages must be similar. This could be a part of the CAB-MSD specification.

3.4 TAP Service Requirements

This has not been discussed in Paris, but the question of the model mapping in TAP responses is relevant.

- *[T1]* CAB-MSD must be designed in such a way that the TAP_SCHEMA can possibly be annotated.

- *[T2]* The TAP_SCHEMA annotation must be designed in a way that the TAP server is capable to generate correct annotations in basic query responses.

4 Proposal

4.1 Overview

As we have discarded the idea of building a global source model, we focus on a consistent and flexible way to provide a complete and homogeneous description of individual quantities (position, velocity ...). This is somehow similar to what has already been adopted for specific cases with GROUPS and UTypes or with specific XML elements (COOSYS, TIMESYS). However, we believe that we can overcome most of the limitations of this actual approach:

- **Lack of flexibility**: The support of new types of quantities may require either standard updates as it happened for VOTable 1.4 with TYMESYS, or code update to support new GROUP/Utype blocks.

- **Lack of homogeneity**: TIMESYS and COOSYS have their own syntax, though similar. GROUP/Utype blocks refer to specific models which are not machine readable. It is furthermore difficult to represent class hierarchies that way; in fact they are rather used to represent flat data.

- **Poor link support**: The lack of a clean way saying what is retrieved by URLs has already been pointed out very often.

- **Poor support for data association:** There is no standard way to join data in a VOTABLE. It is for instance not possible to tell a client that this table contains all the detections of that source or that that another table contains a mix of photometric points with different filters.

Despite these limitations, these solutions are widely used and are satisfying. This is why CAB-MSD is not reinventing the wheel. It aims at doing the same job but by integrating IVOA standards.

- **STC2** (Dittmar and Rots, 2019): The Meas/Coords models provides a uniform representation of most of the used measurements with their frames. STC2 measurements can easily be extended furthermore.

- **PhotDM**(Salgado and Osuna et al., 2013): Provides an accurate description of the photometric filters.

- **VO-DML** (Lemson and Laurino et al., 2018): This is a consistent XML scheme for representing VO models. Thanks to VO-DML, models are machine readable and their components are easily identifiable.

- **VO-DML mapping** (Lemson and Laurino et al., 2017): This is not a standard yet, but a large part of the proposal can be reused to map CAB-MSD in VOTables.

- **Semantic**: The semantic achievements in others context (e.g. datalink Dowler and Bonnarel et al. (2015)) could be reused to qualify CAB-MSD links.

- **Registry**: The registry schema allows to clearly identify models or services referenced by CAB-MSD instances.

- **Instance of other models** (e.g. Provenance, DatasetMetadata) could also be bound with CAB-MSD instance.

Figure 1 illustrates a data workflow using CAB-MSD in 2 steps: 1) The data provider implements a DAL service and annotates all quantities of interest in the result VOTable, 2) The annotated VOTable is downloaded by

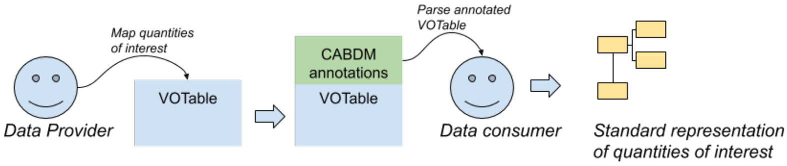

Figure 1: CAB-MSD workflow

the client which can extract the quantities of interest thanks to a CAB-MSD compliant API.

The key point is that the model remains the same whatever the mapped quantities are. The definition of what are the quantities of interest depends on the DAL context. We can imagine that the DAL server annotates anything it can. This would be the case for TAP services. From another hand, it could just annotate a limited set of quantities relevant for a peculiar scientific context. This would be the case for services delivering mission-specific data or feeding specific tools such as time series viewers. In fact, CAB-MSD is more a container than a model. The nature of the quantities carried by the VOTable is discovered by the client. The model just ensures that those quantities are understandable for any CAB-MSD compliant API.

4.2 Model

The class diagram of figure 2 shows the coarse components of CAB-MSD.

- A CAB-MSD instance can contains a set of measurements
  - The set of CAB-MSD measurements is the same for all instances of a given dataset (VOTable).
  - The set of measurements is not defined by the model but it is specific to one VOTable
  - All supported measurements are modeled in CAB-MSD or taken out from imported models.
- A CAB-MSD instance can contains a set of associated data.
  - Other CAB-MSD instances
  - VO products (e.g. SSLAP response)
  - VO services (e.g. Datalinks)
  - Other VO model instances (e.g. Provenance, DatasetMetaData)
  - All links have a semantic tag.

Table 5 gives a few details on the model classes.

We expect that the use of abstract super-classes will help to build mapping snippets that make easier the annotation job for non model experts.

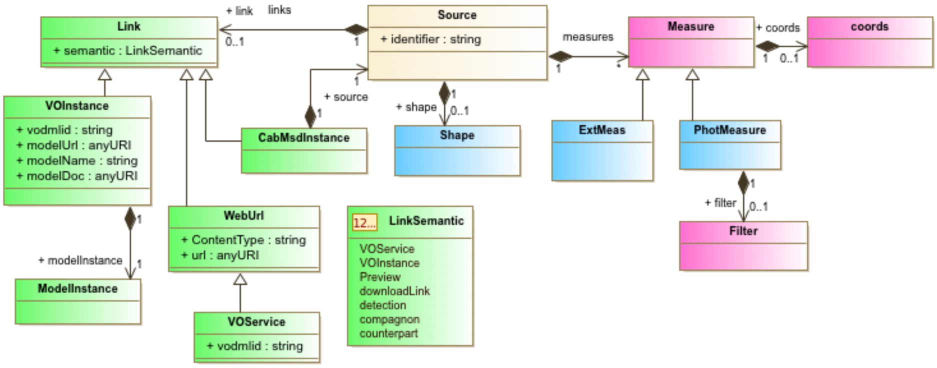

Figure 2: Class diagram (red: imported VO models, blue: CAB-MSD classes, green: associated data)

Class	Role
Source	Root class of the model - just needs a unique identifier
Measure	Imported from STC2 (Time, Position, Velocity, Polarization, Generic Measurment)
PhotometricMeas	Derived from STC2:Meas - connected with PhotDM
ExtMeas	STC2:Meas extension - Describes usual quantities that are not in STC2
Shape	Source shape - The way to put shapes in CAB- MSD is not given here
Links	Superclass for links - just contains the semantic of the link
VOInstance	Link to an instance of another VO models (e.g. Provenance)
ModelInstance	Container for instances of another VO models (TBD))
WebUrl	Pointer to a Web service
VOService	Pointer to a VO compliant Web service
Cabmsd	Pointer to a CAB-MSD instance

Table 5: Main classes of the model

4.3 Annotations

To fulfill the requirements, CAB-MSD standard should embed its own annotation syntax.

This proposal is derived from the VO-DML mapping syntax proposed by (Lemson and Laurino et al., 2017):

- Annotation block separated from the data. Original data are not altered by the mapping, the mapping can work (almost) whatever the way data are organized.
- One annotation block per <TABLE>
- Internal reference based on VO-DML identifiers and on VOTable identifiers.
- Foreign key mechanism

After some tests with time domain data, Michel (2018c) Michel (2018a) proposed enhancements targeting a better human-readability and an easier job for legacy clients and for legacy data curators:

- Liter syntax anywhere when possible
- Usage in some places of element attributes instead of specific XML elements to facilitate the templating
- Usage of XML elements guiding the parser

This work could be a solid basis for the CAB-MSD annotation mechanism.

4.4 Client API

APIs are not part of the standard, but we could benefit from a common definition of browsing CAB-MSD instances. This would facilitate the work for developers and improve the interoperability in sense that new features would have to be first applied to a generic interface before to be coded. Such an interface has been tested with GAIA time series (Michel, 2018b). It is based on selectors using VO-DML identifiers and returning dictionaries (hash maps) rather than (java or Python) class instances. The Java snippet below, extracting the first photometric point of a time series, comes from this demonstrator.

```
/*
 * Extract the first photometric point
 */
MappingElement firstPoint = pointList.getContentElement(0);
List<MappingElement> mesures =
    firstPoint .getSubelementsByRole("meas:CoordMeasure.coord");
/*
 * Measurments of a photometric point are identified by their roles.
 * To extract time and mag, we have to read all of them and to check the roles
 */
```

```
for( MappingElement mes: mesures){
        MappingElement x;
        if( (x = mes.getContentElement("coords:domain.time.JD.date")) != null ) {
                sparseCubeReport.firstTime = x.getStringValue();
        } else if ( (x = mes.getContentElement("ts:Magnitude.value")) != null ) {
            sparseCubeReport.firstMag = x.getStringValue();
        }
}
```

5 Conclusions and Prospects

This note is not detailed enough to ensure that CAB-MSD does fulfill all requirements; table 6 show hows this could be achieved by this proposal.

Minimizing the data annotation cost	A mapping syntax designed to promote the use of templates Only the requested quantities are mapped.
Flexibility	The APIS is based on selectors using VO-DML identifiers (strings passed as function parameters). The code has just to build maps with data matching the selector: no model specific code
Homogeneity	All quantities are mapped with the same syntax. The measurement structure (value/coord frame) is always based on STC.
Link support	Links with different sorts of datasets or services are well supported.

Table 6: Key points of CAB-SMD

A large part of the requirements have been issued from the Paris Interop, but the proposed solutions relies on ideas discussed first in the frame of the VO-DML mapping focus group and then within the Time Series group. They have been tested and code has been published. This is why we are confident to have a solid basis for a model for source data. Paying attention to the interest of the community for CAB-MSD, we need now to check, keyboard on the table, that the use cases are really supported by the model, that the annotation process is acceptable and that coding parsers is easy, especially with AstroPy. This is a big job, thus contributors are welcome. We think it's worth it.

A TAP response annotation

A.1 Storing CAB-MSD instance in a TAP service

The relational mapping of CAB-MSD is quite straightforward. We have one master table for the Source class and one table for each set of associated data (e.g. companions). If must be possible to provide the TAP server with the VO-DML identifiers attached to all of these tables. This can be done with either the TAP_SCHEMA or with some external resources. This information can then be used by the server to set the mapping elements necessary to annotate the response. This should work as long as there is no joint tables, but this discussion is out of the scope of the note.

B A proposal to query CAB-MSD instances in one shot

ADQL is a tabular oriented query language, thus there is no way to use it to get a set of model components in one shot. This can be worked around whether the TAP server is informed that the request concerns model instances. It can then trigger right actions. This notification could done by a specific statement in the SELECT clause. Below is an example of such a hack:

SELECT ivoa.cabmsd **FROM** master_table **WHERE** . . .

This hides the requested columns and thus makes sure that all model quantities will be returned. Once the server has understood that CAB-MSD instances are requested, it can run first the query on the master_table (Source in the model) and then on all the joined tables. All result tables can be stored in the resulting VOTable and properly mapped by using the models tags from either the TAP_SCHEMA or the inner mapping template.

This could work in a simple way as long as the WHERE clause is restricted to master_table.

References

Bradner, S. (1997), 'Key words for use in RFCs to indicate requirement levels', RFC 2119.
`http://www.ietf.org/rfc/rfc2119.txt`

Demleitner, M., Ochsenbein, F., McDowell, J. and Rots, A. (2016), 'Referencing stc in votable', Note 2010-06-18.
`http://ivoa.net/documents/Notes/VOTableSTC/`

Derriere, S. (2016), 'Referencing stc in votable', Note 2011-05-12.
`https://wiki.ivoa.net/internal/IVOA/PhotometryDataModel/NOTE-PPDMDesc-0.2-20110512.pdf`

Dittmar, M. C. and Rots, A. (2019), 'Stc version 2.0', Wiki Page 2016-09-16.
https://wiki.ivoa.net/twiki/bin/view/IVOA/STC

Dowler, P., Bonnarel, F., Michel, L. and Demleitner, M. (2015), 'IVOA datalink', IVOA Recommendation 17 June 2015.
http://www.ivoa.net/documents/DataLink/

Lemson, G., Laurino, O., Bourges, L., Cresitello-Dittmar, M., Demleitner, M., Donaldson, T., Dowler, P., Graham, M., Gray, N., Michel, L. and Salgado, J. (2018), VO-DML: a consistent modeling language for IVOA data models Version 1.0, Technical report.
http://doi.org/10.5479/ADS/bib/2018ivoa.spec.0910L

Lemson, G., Laurino, O., Donaldson, T., Cresitello-Dittmar, M., Michel, L., Dowler, P., Demleitner, M., Graham, M. and Salgado, J. (2017), 'Mapping data models to votable', WD 20170323.
https://volute.g-vo.org/svn/trunk/projects/dm/vo-dml-mapping/doc/VO-DML_mapping_WD.pdf

Michel, L. (2017), 'Mapping time domain data in votables', Talk 2017-07-12.
https://wiki.ivoa.net/internal/IVOA/InterOpOct2017TDIG/TimeDomainMapping.pdf

Michel, L. (2018a), 'Annotating gaia time series with vo-dml', Talk 2018-11-01.
https://wiki.ivoa.net/internal/IVOA/InterOpNov2018TDIG/VODML_Lite_TS.pdf

Michel, L. (2018b), 'Reading vo-dml annotations with java code', Talk 2018-11-01.
https://wiki.ivoa.net/internal/IVOA/InterOpNov2018Apps/VODML_Lite_JAVA_API.pdf

Michel, L. (2018c), 'Serializing time domain data a (simplified) vo-dml approach', Talk 2018-05-10.
https://wiki.ivoa.net/internal/IVOA/InterOpMay2018TDIG/Talk_VODML_Lite_at_a_glance_en.pdf

Osuna, P. et al. (2006), 'Catalog data model', Catalog Data Model Work Package.
https://wiki.ivoa.net/twiki/bin/view/IVOA/IVAODMCatalogsWP

Salgado, J., Lemson, G. and Demleitner, M. (2016), 'Source dm', Source DM 2016-05-10.
https://wiki.ivoa.net/twiki/bin/view/IVOA/SourceDataModel

Salgado, J., Osuna, P., Rodrigo, C., Allen, M., Louys, M., McDowell, J., Baines, D., Maiz Apellaniz, J., Hatziminaoglou, E., Derriere, S. and Lemson, G. (2013), 'IVOA Photometry Data Model Version 1.0', IVOA Recommendation 05 October 2013, arXiv:1402.4752.
http://doi.org/10.5479/ADS/bib/2013ivoa.spec.1005S