International
Virtual
Observatory
Alliance

VO-DML Tools

Version 1.0

IVOA Note 2025-11-11

Working Group
Data Models
This version
https://www.ivoa.net/documents/Notes/VodmlTools/20251111
Latest version
https://www.ivoa.net/documents/Notes/VodmlTools
Previous versions
This is the first public release
Author(s)
Paul Harrison
Editor(s)

Version Control
Revision b00fcde-dirty, 2025-10-24 15:52:35 +0100

Abstract

The IVOA standard modelling language VO-DML Lemson and Laurino et al. (2018) was developed with an associated set of tooling that could be used in the creation of data models expressed in VO-DML. The VO-DML standard does not go into any detail about the tooling as it describes only the language itself. This note attempts to fill that gap by describing the development of the tooling, and give an outline of its functionality. In addition this document promotes the idea that the VO-DML tools are the de-facto standard way to create the schemas for serializations of model instances.

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 in the IVOA document repository[1].

Contents

Acknowledgments

Although the author is the current lead maintainer of the VO-DML Tools, the tools are based on the original contributions principally of Gerard Lemson and Laurent Bourgès, and presumably associated contributions from others working on SIM-DM (Lemson and Wozniak et al., 2012) and the earlier VO-URP[2] project.

1 Introduction

VO-DML Lemson and Laurino et al. (2018) was developed as a machine-readable language for creating data models. This was done to promote re-use and rigour in the model definitions, in contrast to the existing practice

[1] https://www.ivoa.net/documents/

[2] https://github.com/glemson/vo-urp

of defining models only by description in documentation, which allowed an unconstrained set of possibilities, although it was common at least to use UML *UML — Unified Modeling Language* (n.d.). There was a desire to continue to use UML tools to create data models which meant that there was a requirement for some tooling to convert the UML representation to VO-DML - this requirement was met by the development of the VO-DML tools. Although the VO-DML tools were not described in the VO-DML standard as it is possible to create VO-DML definitions 'by hand' in a text editor, this is increasingly infeasible to do as the models grow in complexity - at a minimum there needs to be a mechanism to ensure that any models created do not contain errors and are valid against the VO-DML design criteria. One of the earliest functionalities of the VO-DML tools was to produce standard documentation and diagrams of the data models, which again meant that, in practical terms, the use of the VO-DML tools was a requirement when developing a data model, as the production of such documentation was a requirement of the VO-DML standard.

In addition, one of the main purposes of creating a machine-readable definition of a data model is to be able to then generate products from this representation. However, it is at this point that the VO-DML standard does not make any normative requirements - there is some discussion of how this might be done in in Appendix B of various serialization formats, but not described in sufficient detail to produce interoperable formats. One of the main aims of this document is to propose that the implementation of these serializations that has been done in the VO-DML tools is the de-facto standard for how such concrete serializations are created from VO-DML models.

2 A Brief History of VO-DML Tools[3]

As mentioned in the acknowledgments, the VO-DML tools have their origins in the Simulation Data Model and VO-URP projects, which included Java code generation from a precursor meta-language that was similar to VO-DML. In the process of standardizing the VO-DML meta-language, some capabilities of the precursor language were dropped

-

3 Technologies used to write the Tools

[3] according to the author

4 Using the VO-DML tools

This document is deliberately deficient in detail on exactly how to use the VO-DML tools. This has been done because the intention is that this detail of the use of the tools will be maintained as on-line documentation at https://ivoa.github.io/vo-dml/ rather than published via the usual IVOA document publishing route.

4.1 Features

5 Standardization

6 Future Directions

It should be noted that the VO-DML tools are in a state of active development and have not reached what the author would consider a 1.0 status. The development is controlled by using the usual management mechanisms available in GitHub - in particular issues are created for suggested enhancements and bug fixes and code is controlled via pull requests.

The development of the VO-DML Tools also is a test-bench for future enhancements to the VO-DML language and standard, as new developments will always be prototyped in the tools before being suggested for the the standardization track.

A Changes from Previous Versions

No previous versions yet.

References

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', IVOA Recommendation 10 September 2018. doi:10.5479/ADS/bib/2018ivoa.spec.0910L, https://ui.adsabs.harvard.edu/abs/2018ivoa.spec.0910L.

Lemson, G., Wozniak, H., Bourges, L., Cervino, M., Gheller, C., Gray, N., LePetit, F., Louys, M., Ooghe, B. and Wagner, R. (2012), 'Simulation Data Model Version 1.0', IVOA Recommendation 03 May 2012, arXiv:1402.4744. doi:10.5479/ADS/bib/2012ivoa.spec.0503L, https://ui.adsabs.harvard.edu/abs/2012ivoa.spec.0503L.

UML — Unified Modeling Language (n.d.). https://www.omg.org/spec/UML/.