International
Virtual
Observatory
Alliance

Model Instances in Votables
Version 1.0

IVOA Working Draft 2020-09-15

Working group
DM
This version
http://www.ivoa.net/documents/model-instance-in-vot/20200915
Latest version
http://www.ivoa.net/documents/model-instance-in-vot
Previous versions
This is the first public release
Author(s)
François Bonnarel, Gilles Landais, Laurent Michel, Jesus Salgado, Gerard Lemson
Editor(s)
Laurent Michel, Mark Cresitello Dittmar

Abstract

Vodml-instance-vot proposes a syntax to map VOTable data on any model serialized in VO-DML. Vodml-instance-vot annotations are grouped in a single XML block located in the VOTable head. The annotation block allows to easily reconstruct the model structure. It designed in a way that the block can be reused on different data sets in order to facilitate the annotation process. Vodml-instance-vot is enable to join data from different tables

Status of this document

This is an IVOA Working Draft for review by IVOA members and other interested parties. It is a draft document and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use IVOA Working Drafts as reference materials or to cite them as other than "work in progress".

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

Contents

Acknowledgments

CDS/TDIG/SourceDM contributors

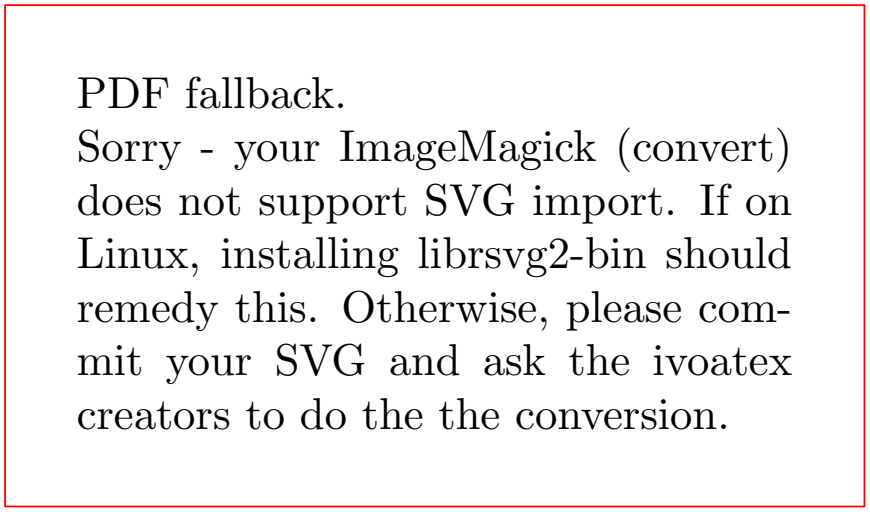


Figure 1: Architecture diagram for this document

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 (**?**).

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

1.1 Role within the VO Architecture

Fig. 1 shows the role this document plays within the IVOA architecture (**?**).

???? and so on, LaTeX as you know and love it. ????

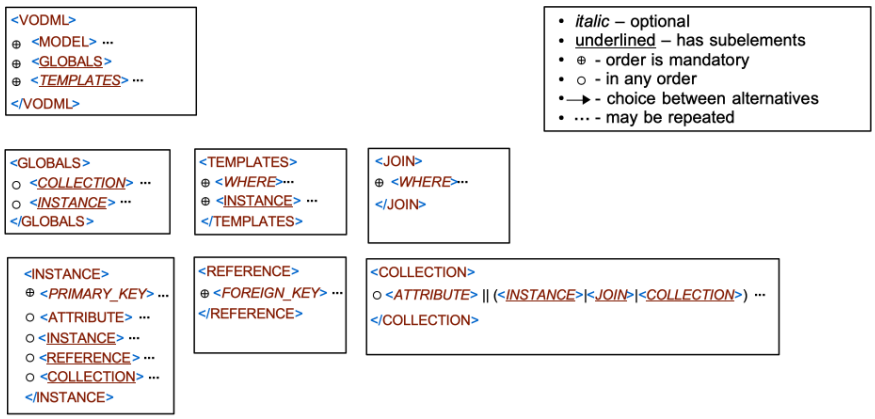
2 Use Cases and Requirements

2.1 Use Cases

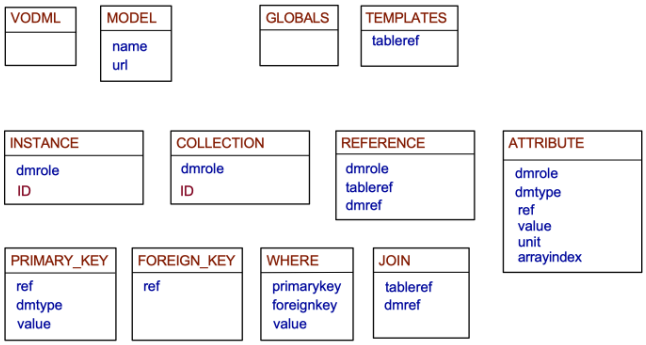
2.2 Requirements

3 Syntax

Element Hierarchy



Attribute Summary



3.1 Mapping Block Scope

The mapping block must be isolated in a VOTable RESOURCE with `type="meta"`. This RESOURCE must be the first child of a RESOURCE `type="results"`. The scope of the mapping block is the whole content of that resource.

The `dm-mapping` name space isolate VOTable elements from mapping elements.

A VOTable RESOURCE type="results" must contain at most one mapping block.

<VOTABLE xmlns="http://www.ivoa.net/xml/VOTable/v1.3"

```
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            version="1.3">
  <RESOURCE type="results">
    <RESOURCE type="meta">
      <dm-mapping:VODML
       xmlns:dm-mapping="http://www.ivoa.net/xml/merged-syntax">
        <dm-mapping:MODEL> ... <dm-mapping:/MODEL>
        <dm-mapping:MODEL> ... <dm-mapping:/MODEL>
        ...
        <dm-mapping:GLOBALS> ... </dm-mapping:GLOBALS>
        <dm-mapping:TEMPLATES> ... </dm-mapping:TEMPLATES>
        <dm-mapping:TEMPLATES> ... </dm-mapping:TEMPLATES>
        ...
      </dm-mapping:VODML>
    </RESOURCE>
    <RESOURCE type="results">
      <TABLE name="Results">
      ....
      </TABLE>
    </RESOURCE>
  </RESOURCE>
</VOTABLE>
```

Listing 1: Mapping block in a VOTable

3.2 Mapping Block Structure

```
  <dm-mapping:VODML>
      <dm-mapping:MODEL> ... <dm-mapping:/MODEL>
      <dm-mapping:GLOBALS> ... </dm-mapping:GLOBALS>
      <dm-mapping:TEMPLATES> ... </dm-mapping:TEMPLATES>
      . ..
  </dm-mapping:VODML>
```

Listing 2: Complete mapping block example

3.3 Syntax

3.3.1 VODML

Top level mapping element.

Element	Position	Cardinality
`MODEL`	1	1-*
`GLOBALS`	2	0-*
`TEMPLATES`	3	0-*

Table 1: Allowed children for `VODLM`

3.3.2 MODEL

A VOTable can provide serializations for an arbitrary number of data model types. In order to declare which models are represented in the file, data providers must declare them through the MODEL elements. Only models that are used in the file must be declared. A model is used if at least one element in the mapping block refer to it. In other terms, only models that define vodml-ids used in the annotation must be declared.

Attribute	Role
@name	Name of the mapped model (informal). This attribute cannot be left empty
@url	Url of the vo-dml serialization of the model. This attribute cannot be left empty if present.

Table 2: `MODEL` attributes

@name	@url	Pattern
MAND	OPT	Unique attribute pattern supported by `MODEL`

Table 3: Valid attribute patterns for `MODEL`

3.3.3 GLOBALS

Some annotations may map the Resource contents to instances or collections of data model types that are global in the mapping scope, possibly because such instances are refer- enced by other intances that annotate specific tables. More generally, some annotations will define instances that are completely defined in terms of constant value, i.e. they are not represented in tabular form. Rather, they are completely and directly represented by an XML element. Such instances should be included in the GLOBALS element. GLOBALS must only contain direct representations of instances, i.e. INSTANCE elements that do not refer to any FIELD directly. This rule is not enforces via the XSD schema Also, GLOBALS should not contain any INSTANCEs with REFERENCEs to indirect INSTANCEs.

Element	Position	Cardinality
`INSTANCE` Any	0-*	
`COLLECTION`	Any	0-*

Table 4: Allowed children for `GLOBALS`

3.3.4 TEMPLATES

Attribute	Role
@tableref	ID of the mapped table.

Table 5: `TEMPLATES` attributes

@tableref	Pattern
OPT	If @tableref is not present, `TEMPLATES` maps the first `TABLE` of the `RESOURCE`

Table 6: Valid attribute patterns for `TEMPLATES`

Element	Position	Cardinality	
`WHERE`	1	0-*	The mapping must be applied to the rows matching the `WHERE` condition only
`INSTANCE`	2	0-*	Mapped class instances

Table 7: Allowed children for `TEMPLATES`

3.3.5 COLLECTION

Attribute	Role
@ID	ID of the `COLLECTION` element, must be unique within the mapping block. Must not be left empty.
@dmrole	Role of the collection in the DM. Must not be empty if present

Table 8: `COLLECTION` attributes

@ID	@dmrole	Pattern
MAND	NO	The collection, usually located in `GLOBALS`, has no role. It can be referenced by a `REFERENCE`
OPT	MAND	The element maps a collection playing a role in the model. @ID can also be set in that case.

Table 9: Valid attribute patterns for `COLLECTION`

Element	Position	Cardinality	
REFERENCE	Any	0-*	Collection item as a reference to either a class or a collection instance. No JOIN allowed if present
INSTANCE	Any	0-*	Collection item as a class instance. No JOIN allowed if present
ATTRIBUTE	Any	0-*	Collection item as a simple attribute. No JOIN allowed if present
COLLECTION	Any	0-*	Collection item as a collection. No JOIN allowed if present
JOIN	1	0-1	The COLLECTION must be populated by a join operation. In this case JOIN must be the unique child of COLLECTION

Table 10: Allowed children for COLLECTION

3.3.6 INSTANCE

VO-DML structured types are annotated by using the INSTANCE element. Note that there is no difference, from a schema point of view, between **ObjectType**s and **DataType**.

Attribute	Role
@ID	ID of the mapping element
@dmrole	Role of the instance in the DM
@dmtype	Class name

Table 11: INSTANCE attributes

@ID	@dmrole	@dmtype	Pattern
MAND	NO	MAND	The instance, usually located in GLOBALS, has no role. It can be referenced by a REFERENCE
OPT	MAND	MAND	The element maps a instance playing a role in the model. @ID can also be set in that case.

Table 12: Valid attribute patterns for INSTANCE

Element	Position	Cardinality	
REFERENCE	Any	0-*	Object attribute as a reference to either a class or a collection instance.
INSTANCE	Any	0-*	Object attribute as a class instance.
ATTRIBUTE	Any	0-*	Object attribute as a simple attribute.
COLLECTION	Any	0-*	Object attribute as a collection.

Table 13: Allowed children for INSTANCE

3.3.7 ATTRIBUTE

Attribute	Role
@dmrole	Role of the attribute in the DM
@dmtype	Type of the attribute in the DM
@ref	Reference of the FIELD or PARAM that has to be sued to set the ATTRIBUTE value.
@value	Default ATTRIBUTE value. This value is taken if there is no @ref attribue or if @ref cannot be resolved.
@unit	ATTRIBUTE unit. This is the unit in which the native value must be converted to be complient with the model. This attribute is always optional.
@arrayindex	Index of the native value to be taken to set the ATTRIBUTE. Must be ignored if the native value is a single value. An error must be risen if @arrayindex is out of range.This attribute is always optional.

Table 14: ATTRIBUTE attributes

@dmrole	@dmtype	@ref	@value	Pattern
MAND	MAND	MAND	OPT	The ATTRIBUTE value must be set with the value of the element referenced by @ref. The @ref can not be resolved and @value is present, @value must taken as ATTRIBUTE value
MAND	MAND	NO	MAND	The ATTRIBUTE value must be set with @value

Table 15: Valid attribute patterns for ATTRIBUTE

3.3.8 REFERENCE

Complex pattern that must be detailed later in a specific section

Attribute	Role
@dmrole	Role of the referenced instance or collection in the DM
@tableref	ID of the `COLLECTION` to be joined with in case of using a `FOREIGN_KEY`
@dmref	ID of the referenced instance or collection

Table 16: `REFERENCE` attributes

@dmrole	@tableref	@dmref	Pattern
MAND	MAND	NO	This is the `FOREIGN_KEY` pattern. @tableref gives the ID of the `COLLECTION` to be joined with. In this case `REFERENCE` must have one `FOREIGN_KEY` child and the joined `COLLECTION` must have a `PRIMARY_KEY`
MAND	NO	MAND	Simple reference to either an `INSTANCE` or `COLLECTION`, usually searched in the `GLOBALS`

Table 17: Valid attribute patterns for `REFERENCE`

3.3.9 JOIN

Attribute	Role
@tableref	Reference of the table to be joined with.
@dmref	Reference of the `COLLECTION` (in `GLOBALS` to be joined with.

Table 18: `JOIN` attributes

@tableref	@dmref	Pattern
MAND	NO	The join is done against the table identified by @tableref
NO	MAND	The join is done against the `COLLECTION` identified by @rmref

Table 19: Valid attribute patterns for `JOIN`

Element	Position	Cardinality	
WHERE	1	0-*	Join condition

Table 20: Allowed children for JOIN

3.3.10 WHERE

Attribute	Role
@primarykey	FIELD identifier of the primary key column
@foreignkey	FIELD identifier of the foreign key column
@value	Literal value the @primarykey cell must match with

Table 21: WHERE attributes

@primarykey	@foreignkey	@value	Pattern
MAND	MAND	NO	2 tables join criteria: @primarykey = @foreignkey
MAND	NO	MAND	Simple join criteria: @primarykey = @value

Table 22: Valid attribute patterns for WHERE

3.3.11 PRIMARY_KEY

Attribute	Role
@ref	ID of the FIELD used as primary key
@dmtype	Type of the key
@value	Literal key value. Used when the key relates to a COLLECTION in the GLOBALS

Table 23: PRIMARY_KEY attributes

@ref	@dmtype	@value	Pattern
MAND	MAND	NO	The FIELD referenced by @ref is a primary key. This pattern is used within a TEMPLATES
NO	MAND	MAND	@value gives the key value. This pattern is used to set a primary key to a COLLECTION

Table 24: Valid attribute patterns for PRIMARY_KEY

3.3.12 FOREIGN_KEY

Attribute	Role
@ref	Only used in `REFERENCE`. Identifier of the FIELD that must match the primary key of the referenced collection

Table 25: `FOREIGN_KEY` attributes

4 Changes from Previous Versions

No previous versions yet.